NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SELECTING A SOFTWARE CAPABILITY EVALUATION FOR WEAPONS ACQUISITION

by

Ferdinand M. Raguindin

September 1994

Principal Advisor: Martin McCaffrey
Associate Advisor: James Emery

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1994.

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE SELECTING A SOFTWARE CAPABILITY EVALUATION FOR WEAPONS ACQUISITION (U)

5. FUNDING NUMBERS

6. AUTHOR(S) Raguindin, Ferdinand M.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE
A

13. ABSTRACT *(maximum 200 words)*

This research reviews the two most widely used software capability evaluations in the Department of Defense: the Software Capability Evaluation (SCE) created by the Software Engineering Institute (SEI), and the Software Development Capability Evaluation (SDCE) created by the US Air Force's Aeronautic Systems Command (ASC). Their use as part of the source selection evaluation of a contractor developing a software intensive weapon system is examined. The objective of this thesis is to describe each evaluation method, then highlight their respective strengths and weaknesses. The result is a guide that will assist Program Managers in deciding which software capability evaluation is more suitable for use in their program.

14. SUBJECT TERMS
Software Engineering, Software Capability Evaluation, Systems Acquisition

15. NUMBER OF PAGES
141

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

SELECTING A SOFTWARE CAPABILITY EVALUATION FOR WEAPONS ACQUISITION

by

Ferdinand M. Raguindin
Captain, United States Army
B.A., Georgetown University, 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1994

Author: Ferdinand M. Raguindin
Ferdinand M. Raguindin

Approved by: Martin J. McCaffrey
Martin McCaffrey, Principal Advisor

James Emery
James Emery, Associate Advisor

David Whipple, Chairman
Department of Systems Management

ABSTRACT

This research reviews the two most widely used software capability evaluations in the Department of Defense: the Software Capability Evaluation (SCE) created by the Software Engineering Institute (SEI), and the Software Development Capability Evaluation (SDCE) created by the US Air Force's Aeronautic Systems Command (ASC). Their use as part of the source selection evaluation of a contractor developing a software intensive weapon system is examined. The objective of this thesis is to describe each evaluation method, then highlight their respective strengths and weaknesses. The result is a guide that will assist Program Managers in deciding which software capability evaluation is more suitable for use in their program.

TABLE OF CONTENTS

I. INTRODUCTION 1

A. BACKGROUND 1

B. OBJECTIVES 4

C. THE RESEARCH QUESTION 4

D. SCOPE, LIMITATIONS, AND ASSUMPTIONS 5

E. LITERATURE REVIEW AND METHODOLOGY 6

F. ORGANIZATION OF STUDY 7

II. THE CAPABILITY MATURITY MODEL 8

A. DEVELOPMENT 8

B. STRUCTURE 10

C. MATURITY LEVELS 12

1. Level 1: Initial 12

2. Level 2: Repeatable 13

3. Level 3: Defined 16

4. Level 4: Managed 20

5. Level 5: Optimizing 22

D. PLANNED IMPROVEMENTS 24

E. SUMMARY 25

III. THE SOFTWARE CAPABILITY EVALUATION 26

A. DEVELOPMENT 27

B. CRITICISMS AND CHANGES 27

C. SCE VERSION 1.5 29

1. Phase 1: Evaluation Start 31

2. Phase 2: General Preparation 35

3. Phase 3: Specific Preparation 36

4. Phase 4: Site Data Collection (Site Visit) 38

5. Phase 5: Findings 41

D. SUMMARY 43

IV. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION MODEL 44

A. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION METHOD 44

B. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION MODEL 47

C. MODEL DEVELOPMENT 47

D. MODEL STRUCTURE 48

E. SOFTWARE DEVELOPMENT CAPABILITY EVALUATION MODEL FUNCTIONAL AREA OVERVIEW 51

1. Functional Area 1: Program Management 51

2. Functional Area 2: Systems Engineering 54

3. Functional Area 3: Software Engineering 55

4. Functional Area 4: Quality Management and Product Control . . 56

5. Functional Area 5: Organization Resources and Program Support 58

6. Functional Area 6: Program Specific Technologies 59

F. SUMMARY 60

V. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION ACTIVITIES 61

A. ACTIVITY A: DETERMINE APPLICABILITY 61

B. ACTIVITY B: SELECT AND PREPARE TEAM 65

C. ACTIVITY C: PREPARE PLAN AND SCHEDULE 68

D. ACTIVITY D: TAILOR SOFTWARE DEVELOPMENT CAPABILITY EVALUATION, SELECT CRITERIA AND QUESTIONS 69

E. ACTIVITY E: PREPARE REQUEST FOR PROPOSAL AND INSTRUCTIONS 70

F. ACTIVITY F: REVIEW PROPOSALS 72

G. ACTIVITY G: PLAN FOR AND CONDUCT SITE VISIT 72

H. ACTIVITY H: ANALYZE CLARIFICATION REQUESTS AND DEFICIENCY REPORTS 74

I. ACTIVITY I: EVALUATE, SCORE, AND INTEGRATE RESULTS INTO SOURCE SELECTION 74

J. ACTIVITY J: INCORPORATE INTO CONTRACT 77

K. ACTIVITY K: CONCLUDE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION TEAM ACTIVITIES 77

L. ACTIVITY L: CONDUCT FORMAL FEEDBACK 77

M. ACTIVITY M: SUPPORT PROGRAM FOLLOW-THROUGH 78

N. SUMMARY 78

VI. LITERATURE RESEARCH 79

A. AEROSPACE CORPORATION STUDY 79

B. INSTITUTE FOR DEFENSE ANALYSIS (IDA) STUDY 81

C. SUMMARY 84

VII. COMPARISON AND ANALYSIS 85

A. COST OF THE EVALUATIONS 85

1. Issue 85

2. Research Findings 85

3. Conclusions 87

B. SCOPE OF THE SDCE AND SCE 87

1. Issue 87

2. Findings of Previous Studies 88

3. Research Findings 88

4. Conclusion 91

C. CREDIT FOR NEW PROCESSES 92

1. Issue 92

2. Findings of Previous Studies 92

3. Research Findings 92

4. Conclusion 93

D. QUESTIONNAIRES 93

1. Issue 93

2. Findings of Previous Studies 94

3. Research Findings 94

4. Conclusion 97

E. TAILORING 97

1. Issue 97

2. Findings of Previous Studies 98

3. Research Findings 98

4. Conclusions 100

F. PROJECT FOCUS 100

1. Issue 100

2. Findings of Previous Studies 100

3. Research Findings 101

4. Conclusions 101

G. CONDUCTING SITE VISITS 102

1. Issue 102

2. Findings of Previous Studies 102

3. Research Findings 102

4. Conclusion 104

H. PROCESS IMPROVEMENT 105

1. Issue 105

2. Findings of Previous Studies 105

3. Research Findings 105

4. Conclusion 108

I. SAMPLING ERROR 108

1. Issue 108

2. Research Findings 108

3. Conclusions 109

J. DIFFERENCES IN SPA AND SCE RESULTS 110

1. Issue 110

2. Findings of Previous Studies 110

3. Research Findings 110

4. Conclusion 112

K. RECORD OF SUCCESS 112

1. Issue 112

2. Findings of Previous Studies 113

3. Research Findings 114

4. Conclusions 114

L. Summary 115

VIII. SUMMARY AND RECOMMENDATIONS 116

A. ANSWERS TO THE SECONDARY RESEARCH QUESTIONS . . . 116

1. What is the Capability Maturity Model? 116

2. What is a Software Capability Evaluation (SCE) and how is it used during source selection? 116

3. What is the Software Development Capability Evaluation Model? 117

4. What is the Software Development Capability Evaluation and how is it used during source selection? 117

B. THE PRIMARY RESEARCH QUESTION 117

1. SDCE Strengths 118

2. SDCE Weaknesses 119

3. SCE Strengths 121

4. SCE Weaknesses 122

C. RECOMMENDATIONS 123

D. AREAS FOR FURTHER RESEARCH 124

E. SUMMARY 125

LIST OF REFERENCES 126

INITIAL DISTRIBUTION LIST 128

I. INTRODUCTION

A. BACKGROUND

In today's environment of acquiring weapon systems within the Department of Defense (DOD), effective software management is vital to the success of a program. Modern weapon systems rely heavily on their computer hardware and software to operate in a manner in which they were designed [Ref. 1].

Several factors have led to the tremendous growth in the use of computer hardware and software in today's weapon systems. The increasing capabilities of threat weapons, combined with a finite defense budget, led to DOD's policy of emphasizing qualitative rather than quantitative weapons superiority. Advances in microprocessor and integrated circuit technology led to tremendous increases in the capabilities of computer hardware with corresponding decreases in size, power requirements, and costs. As performance requirements for new weapon systems increase, so does the number of on-board computers and the number of source lines of code (SLOC). [Ref. 1:p. 2-2]

The growth in SLOC found in weapons is illustrated by US Air Force (USAF) fighter aircraft. The Vietnam era F-4 Phantom fighter had virtually no software. Today's F-16D Falcon fighter has approximately 236,000 SLOC. The next generation fighter, the Advanced Tactical Fighter (ATF), is projected to require 5,000,000 to 7,000,000 SLOC. [Ref. 2] Computers and software have improved weapon system performance and have expanded the capabilities of the human operator.

While mission-critical software has grown dramatically in complexity and magnitude, the systems and software engineering management discipline necessary to successfully develop this software has not kept pace. Program Managers (PMs) and defense contractors must meet the difficult task of developing large and complex software systems that are critical to system performance often without well-defined and consistently applied systems engineering, software engineering, and management discipline. [Ref. 3] Such shortfalls have adversely impacted many weapon system programs. Software development problems are well-documented in numerous General Accounting Office (GAO) reports [Ref. 4].

The most common software problems are development problems, which lead to program schedule slippages, cost overruns, and delivered software that does not meet the stated requirements [Ref. 1:p. 1-1]. In addition, software production remains labor intensive. Throughout the late 1980s and early 1990, there was increasing concern within DOD and the defense industry that there would not be enough labor available to produce software to meet the increasing demands of both the US Government and the civilian sector. Because of these problems, software has been referred to by some as the "achilles heel" of DOD. [Ref. 2:p. 28]

An important theme in Total Quality Management (TQM) is that the quality of a product is directly related to the process used to produce it. Through continuous process improvement, variance in manufacturing time and cost is reduced. The frequency of meeting customer requirements increases. [Ref. 5] Process improvement normally leads to an increase in productivity [Ref. 6]. For these reasons, attention has

focused on the process used to develop software as a key factor in addressing the problems facing software development today.

The Software Engineering Institute (SEI) defines process capability as the inherent ability of a process to produce planned results [Ref. 7]. Planned results means meeting all requirements within time and cost restraints. In the past, contractors were selected through the source selection process based on factors such as past performance, technical approach, cost, and schedule. An important factor that was ignored during this source selection process was whether prospective vendors had the process capability (in terms of software engineering, systems engineering, and management controls) to execute their respective proposed programs as planned. The challenge facing today's PMs is selecting the best contractor that not only has the capacity but also the capability to produce mission-critical software. [Ref. 3:p. 1]

Capability evaluations are independent evaluations of a vendor's software development process that identify its strengths and weaknesses as they relate to a particular acquisition [Ref. 1:pp. 8-12 - 8-15]. Evaluations can be conducted to determine that a reasonable software process is practiced, documented, enforced, staffed by well-trained personnel, and measured. Capability evaluations have been used in DOD source selection since the late 1980's [Ref. 15:p. 4]. The decision to use evaluations on a project was mainly left to the discretion of the PM.

With the release of USAF Acquisition Policy Memorandum 93M-003 on June 4, 1993, the USAF became the first Service to require the use of capability evaluations during source selection of software intensive projects which meet certain criteria. Two

capability evaluations are authorized for use by the USAF. They are the Software Capability Evaluation (SCE) developed by the SEI, and the Software Development Capability/Capacity Review (SDCCR) created by the Department of the Air Force Aeronautical Systems Center, Air Force Materiel Command (ASC/AFMC).

Since June 4, 1993, both capability evaluations have been revised. In July 1993, SEI released SCE version 1.5. In November 1993, ASC/AFMC published the Software Development Capability Evaluation (SDCE), which supersedes the SDCCR.

B. OBJECTIVES

The objective of this thesis is to provide assistance to PMs in deciding which of the two software capability evaluations, SEI's Software Capability Evaluation version 1.5 or ASC/AFMC's Software Development Capability Evaluation, is better suited for use on their program during source selection for evaluating a contractor's capability to develop a software intensive weapon system. To meet this objective, an overview of each capability evaluation method, together with their corresponding models, will be presented. The strengths and weaknesses of each evaluation will be analyzed. The result will be a guide that a PM can use in selecting between the two software capability evaluation methods.

C. THE RESEARCH QUESTION

The primary research question for this thesis is "What are the strengths and weaknesses of the SEI's Software Capability Evaluation and ASC/AFMC's Software Development Capability Evaluation, that PMs should consider when deciding which

method to use on their program during source selection to evaluate a contractor's software development capability ?" In support of the primary research question, this thesis will address the following subsidiary questions:

1. What is the Capability Maturity Model?

2. What is a Software Capability Evaluation and how is it used during source selection?

3. What is the Software Development Capability Evaluation Model?

4. What is a Software Development Capability Evaluation and how is it used during source selection?

D. SCOPE, LIMITATIONS, AND ASSUMPTIONS

Acquisition Policy Memorandum 93M-003 applies only to the USAF. The USAF is the only Service to use the SDCE and its predecessor, the SDCCR. This thesis, therefore, only examines the USAF's use of the SCE and SDCE during source selection.

The use of capability evaluations is not limited to source selection only. Other uses include monitoring contractor performance after the contract has been awarded, selecting subcontractors by the prime contractor, and as input to selecting the appropriate metrics required to effectively measure the status of a program's software development. This thesis, however, will only address the use of capability evaluations during source selection evaluations of a contractor.

This thesis assumes that the reader is familiar with Government source selection procedures. Discussions concerning source selection organizations, responsibilities,

regulations, and procedures found within this thesis are limited to those areas affected by the capability evaluation process.

E. LITERATURE REVIEW AND METHODOLOGY

The research for this thesis was conducted in two steps. The first steps involved an intensive literary search of published material concerning both capability evaluations. The main effort focused on the written work of each organization on the concept and implementation of their respective evaluation method.

For the Capability Maturity Model (CMM) on which the SCE is based, the primary SEI works are ***CMU/SEI-93-TR-24 Software Engineering Institute, Capability Maturity Model (CMM) Software, Version 1.1*** and ***CMU/SEI-93-TR-25 Software Engineering Institute, Key Practices of the Capability Maturity Model, Version 1.1***. The majority of the information for the SCE itself came from ***CMU/SEI-TR-17 Software Engineering Institute, Software Capability Evaluation (SCE) Version 1.5 Method Description***.

Information concerning the SDCE came primarily from ***AFMC Pamphlet 800-61, Acquisition Management, Software Development Capability Evaluation*** dated 24 November 1993, which is published in two volumes. Other sources of information include articles from professional journals as well as books on software quality.

The second aspect of this research involved conducting interviews to obtain answers to specific questions not addressed in any literature. An interview was conducted with the originator of the USAF's acquisition policy to clarify the contents of the memorandum. Personnel from SEI and ASC/AFMC were interviewed about their

respective evaluation methods. Contractors were also interviewed to get an industry perspective on these evaluations.

F. ORGANIZATION OF STUDY

The remaining chapters of this thesis are organized into three major parts. The first part, consisting of Chapters II through V, will introduce the reader to the two capability evaluations being studied, and the models on which they are based. Chapter II describes the CMM. The next chapter outlines the procedures on how to conduct an SCE. The SDCE model is described in Chapter IV, while the actual SDCE activities are covered in Chapter V. Chapter VI presents the results of two studies.

The second part involves a detailed comparison of both capability evaluations. Chapter VII compares each method in terms of key factors to highlight the strengths and weaknesses of the SCE and SDCE.

Finally, Chapter VIII concludes with a summary of the strengths and weaknesses of each evaluation. This may serve as a guide for PMs in determining which evaluation to use during source selection evaluation for their program.

II. THE CAPABILITY MATURITY MODEL

This chapter introduces the reader to the Capability Maturity Model (CMM) version 1.1. The SCE measures a contractor's strengths, weaknesses, and improvement activities against the principles of the CMM. It is therefore important for the reader to understand the CMM in order to understand the SCE. This chapter provides an overview of the CMM. A more detailed description of the model can be found in the SEI publication, ***CMU/SEI-93-TR-24 Software Engineering Institute, Capability Maturity Model (CMM) Software, Version 1.1.***

A. DEVELOPMENT

The USAF established the SEI in December of 1984, an affiliate of the Carnegie-Mellon University in Pittsburgh, Pennsylvania, under contract as a Federally Funded Research and Development Center (FFRDC). The SEI's mission is to provide leadership in advancing the state of the practice of software engineering to improve the quality of systems that depend on software. [Ref. 8] It was tasked with researching the transition of new software technology, analyzing software development environments, and providing education in the software and system engineering process. [Ref. 1:p. 4-7] SEI's primary vision is to bring engineering and discipline to the development and maintenance of software [Ref. 8]. It is this vision that led to the creation of the CMM.

In November 1986, SEI, with the assistance of the MITRE Corporation, began developing a software process maturity model to create a conceptual structure for improving the management and development of software products in a disciplined and consistent way. SEI released a brief description of the model in September 1987. It was later expanded in Watts Humphrey's book, ***Managing the Software Process***. SEI developed a maturity questionnaire, a set of "yes/no" questions that address organization and management issues, based on this model.

Over the next few years, SEI created two methods for using the questionnaire and the maturity model. The Software Process Assessment (SPA) provides the means for an organization to perform self-audits to identify its strengths, weaknesses, existing improvement activities, and areas for improvement. The second assessment method, the Software Capability Evaluation (SCE), is a tool used by an outside agency to evaluate an organization's software process capability. The results are used during source selection. Feedback on the software process maturity model was incorporated in CMM version 1.0, released in August 1991. The current version of the CMM, version 1.1, was released in February 1993, and is the result of comments from an April 1992 workshop and ongoing feedback from the software community. [Ref. 7;pg. 18-19]

The CMM does not guarantee that software products will be successfully built or that all problems in software engineering will be adequately resolved [Ref. 8:p. 135]. It gives developers a tool to gain control of their software development process and move towards continuous process improvement. It focuses on a set of key process areas that have been proven to enhance software development and maintenance capability. The

CMM does not address every issue concerning the software development process and quality improvement. Issues that are not directly addressed include specific tools, methods and technologies, concurrent engineering and teamwork, systems engineering, human resources, change management, and expertise in a particular domain [Ref. 9]. By focusing on a limited set of activities and working aggressively to achieve them, a developer can steadily improve its process capability [Ref. 8:p. 7].

B. STRUCTURE

The CMM resembles a hierarchial structure. The highest level is the maturity level. SEI defines a maturity level as a well-defined evolutionary plateau on the path towards becoming a mature software organization. There are five maturity levels, with the lowest being Maturity Level 1. Each level serves as a foundation for the subsequent maturity level. [Ref. 7:p. 20]

SEI claims that as an organization's maturity increases, three types of improvements can be expected. First, the difference between planned results and actual results decreases across projects. Second, the variability of actual results around targeted results decreases. Finally, as an organization matures, cost and development time decrease while productivity and product quality increase. SEI admits that there are no definitive studies confirming these claims. These expectations are based on quantitative results of process improvements achieved by organizations using the SPA and the CMM. [Ref. 10:p. 3-54]

With the exception of Maturity Level 1, each maturity level is composed of several key process areas (KPA). KPAs are groups of related activities that, when performed collectively, achieve a set of goals considered important for enhancing process capability. They identify the issues that must be addressed to achieve a specific maturity level. [Ref. 7:p. 26]

Each KPA has a set of goals that specify the KPA's scope, boundaries, and intent. They are used to determine whether an organization has effectively implemented a key process area. [Ref. 7:p. 26]

KPAs are organized into five sections called common features. The common features consist of attributes that indicate whether the implementation and institutionalization of a key process area are effective, repeatable, and lasting. The five common features are:

• Commitment to Perform - Describes actions the organization must take to ensure the process is established and will endure. This typically involves establishing organizational policies and obtaining senior management commitment.

• Ability to Perform - Provides the preconditions that must exist within the organization to implement the software process competently in terms of resources, organizational structure, delegation of responsibility, and training.

• Activities Performed - Conveys the roles and procedures required to implement a KPA. It involves establishing plans and procedures, performing the work, tracking the work, and taking corrective actions as required.

• Measurement and Analysis - Describes the need to measure the process and analyze the results. This typically involves obtaining sample measurements to determine the status and effectiveness of the activities being performed.

• Verifying Implementation - Provides the steps necessary to ensure all activities are performed in compliance with the established process. These steps include reviews and audits conducted by senior management, project management, and software quality assurance personnel. [Ref. 7:p. 26]

The common features contain key practices that describe the infrastructure and activities that contribute most to the effective implementation and institutionalization of a key process area. They consist of a single descriptive sentence, and often include subpractices that describe what should occur for the key process to be implemented satisfactorily. Supplementary information, which contains examples, elaborations, and references to other KPAs, is also provided. While key practices describe what should be done, they do not mandate how to do it. [Ref. 7:p. 26]

C. MATURITY LEVELS

As previously mentioned, maturity levels are a well-defined evolutionary plateau on the path towards becoming a mature software organization. They are used as a means of characterizing an organization's process capability.

1. Level 1: Initial

SEI characterizes this initial level as immature. An immature organization normally does not provide a stable environment for developing and maintaining software.

Processes are ad hoc and often improvised. These organizations frequently have difficulty meeting commitments. This frequently results in a series of crises during which planned procedures, especially testing and reviews, are reduced in scope or abandoned. [Ref. 7:p. 19]

Immature organizations do deliver software that meets all requirements, but they are usually behind schedule and over budget. For Level 1 organizations, success is dependent on the talents and efforts of individuals. Continued success requires using the same development team on new projects or the constant recruitment of highly skilled and competent personnel. Occasionally, good software managers can instill a disciplined approach to software development. However, when they leave their stabilizing influence leaves with them. SEI characterizes Level 1 as unpredictable. The main focus is on the talents of individuals. [Ref. 7:pp. 21-22]

2. Level 2: Repeatable

At the repeatable level, the organization has established basic policies and implementation procedures for managing a software project. Realistic project planning and effective management controls are the result of lessons learned from previous projects. Management tracks costs, schedule, functionality, and problems. Management must initiate and champion the improvement effort. Effective customer-supplier relationships are also established with subcontractors. Only with management discipline can good software engineering practices be retained, especially during periods of crisis and pressure. The goal of Level 2 organizations is to institutionalize management practices shown to be successful in past projects, so that the same success can be repeated

in the future. SEI summarizes the process capability of Level 2 organizations as "disciplined" because project planning and tracking are stable and earlier successes can be repeated. [Ref. 7:p. 22] This level contains six KPAs which are:

• Requirements Management - The purpose is to establish a common understanding of the project requirements between the customer and the development team. This agreement forms the basis for estimating, planning, performing, and tracking the project's software activities. Goals of this KPA are that system requirements allocated to software are controlled to establish a baseline for software engineering and management use. Software plans, products, and activities are kept consistent with the system requirements allocated to software.

[Ref. 8:pp. 55-57]

• Software Project Planning - The purpose is to establish reasonable plans for performing the software engineering and for managing the software project. It involves developing estimates for the work to be performed, establishing the necessary commitments, and defining the plan to perform the work. This plan is necessary for initiating the software effort and managing the work. Its goals include that software estimates are documented for use in planning and tracking the software project, software project activities and commitments that are planned and documented. It is also used for the affected groups and individuals that agree to their commitments related to the software project. [Ref. 8:pp. 58-62]

• Software Project Tracking and Oversight - The purpose is to provide adequate visibility of the actual progress being made so that management can take corrective

actions when performance deviates significantly from the plan. It involves tracking and comparing software accomplishments and results against documented estimates and plans. The plans and estimates are adjusted as necessary. The goals of this KPA are that actual results and performances are tracked against the software plans. Corrective actions are taken and managed to closure when actual results and performance deviate significantly from the software plans. Changes to software commitments are agreed to by the affected groups and individuals. [Ref. 8:pp. 63-64]

• Software Subcontract Management - It provides a means to select qualified software subcontractors and manage them effectively. This involves selecting a software subcontractor, establishing commitments with the subcontractor, and tracking and reviewing the subcontractor's performance and results. "Qualified" does not mean a subcontractor with the "highest process capability." The intent is to find one that is capable of meeting the requirements. This KPA's goals are that the prime contractor selects qualified software subcontractors, the prime contractor and the software subcontractor agree to their commitments to each other, the prime contractor and the software subcontractor maintain ongoing communications, and the prime contractor tracks the software subcontractor's actual results and performance against its commitments. [Ref. 8:pp. 65-67]

• Software Quality Assurance - Its purpose is to provide management with appropriate visibility into the process being used and the products being built. This requires reviewing and auditing the software products and activities to verify they comply with the established procedures and standards. The development team and management

are provided with the results of those reviews and audits. Software quality assurance personnel should be independent of the software producers and project managers. The goals include that software quality assurance activities are planned; software products and activities adhere to applicable standards, procedures, and requirements and are verified objectively; affected groups and individuals are informed of software quality assurance activities and results; and that noncompliance issues that cannot be resolved within the software project are addressed by senior management. [Ref. 8:pp. 68-70]

• Software Configuration Management - Its purpose is to establish and maintain the integrity of the products of the software project throughout the software life cycle. It involves identifying configuration items/units, systematically controlling changes, and maintaining the integrity and traceability of the configuration throughout the software life cycle. The goals are that software configuration management activities are planned; selected software work products are identified, controlled, and available; changes to identified software work products are controlled; and affected groups and individuals are informed of the status and content of software baselines. [Ref. 8:pp. 71-74]

3. Level 3: Defined

At the defined level, the standard process for both management and software engineering activities is documented, standardized, and integrated into what the CMM refers to as the organization's standard software process. It includes inputs, standards, and procedures for performing the work, verification mechanisms such as peer reviews, outputs, and completion criteria. Because the process is well-defined, management has good visibility on the progress of all projects. An organization-wide training program is

established to ensure all personnel possess the knowledge and skill required to implement the standard software process. They are also taught the roles and responsibilities of management and the development team during each phase of the process. Project teams use an approved, tailored version of the organization's standard software process for developing and maintaining software. It takes into account the project's unique characteristics. [Ref. 7:p. 22]

The software process capability of a Level 3 organization is summarized as standard and consistent by the SEI. This is because both software engineering and management activities are stable and repeatable. Cost, schedule, and functionality are under control and quality is tracked. Process capability is based on a common understanding of the activities, roles, and responsibilities in a defined process throughout the organization. [Ref. 7:p. 22] The defined level contains seven KPAs, which include:

• Organization Process Focus - Its purpose is to establish the organizational responsibility for software process activities that improve the organization's overall process capability. It involves developing and maintaining an understanding of the organization standard software process and the tailored project software process. Efforts are also directed at coordinating the activities to assess, develop, maintain, and improve these processes. The typical mechanism to accomplish this is the Software Engineering Process Group (SEPG). Other mechanisms include process review boards, quality circles, and process steering committees. The goals of this KPA are: the software process development and improvement activities are coordinated across the organization, the strengths and weaknesses of the software process used are identified relative to a process

standard, and the organization level process development and improvement activities are planned. [Ref. 8:pp. 79-81]

• Organization Process Definition - This KPA aims to develop and maintain a set of software process assets that improve process performance and provide a basis for cumulative, long term benefits. Software process assets include the organization's standard software process, descriptions of the software life cycles approved for use, the guidelines and criteria for tailoring the organization's standard software process to fit the project's uniqueness, the organization's software process database, and the library of software process related documentation. It involves developing and maintaining the organization's standard software process and related process assets. Its two goals are: standard software process for the organization is developed and maintained, and information related to the use of the organization's standard software process by the software projects is collected, reviewed, and made available. [Ref. 8:pp. 82-89]

• Training Program - The objective is to develop the knowledge and skills of individuals so they can perform their required roles effectively and efficiently. It entails identifying the training needs of the organization and project teams. It also includes developing and/or procuring training to address these needs. Training may include informal as well as formal methods. Training at this level differs from that at Level 2. Unlike Level 3, training at Level 2 is not likely to have been institutionalized across the organization. The goals are that training activities are planned, training for developing the skills and knowledge needed to perform software management and technical roles is

provided, and individuals in the software engineering group and other software related groups receive the training necessary to perform their roles. [Ref. 8:pp. 90-92]

• Integrated Software Management - It is used to integrate the project's software engineering and management activities into a coherent, defined software process tailored from the organization's software process assets. It involves developing the project's defined software process by tailoring the organization's standard software process and managing it accordingly. The goals of Integrated Software Management are that the project's defined software process is a tailored version of the organization's standard software process and the project is planned and managed according to the project's defined software process. [Ref. 8:pp. 93-94]

• Software Product Engineering - Its purpose is to consistently perform a well-defined engineering process that integrates all the software engineering activities to produce correct, consistent software products effectively and efficiently. This requires performing the engineering tasks such as requirements analysis, design, code, and test, and to build and maintain software using appropriate tools and methods. The goals of this KPA are that software engineering tasks are defined, integrated, and consistently performed to produce the software, and secondly, that software products are kept consistent with each other. [Ref. 8:pp. 95-97]

• Intergroup Coordination - This is used to establish a vehicle for the software engineering group to participate actively with the other groups so the project is better able to satisfy the customer's requirements. These other groups may exist within or outside the organization. Examples include systems engineering, marketing, and training. It

involves the disciplined interaction and coordination of the project engineering groups with others to address system level requirements, objectives, and plans. Its goals include that the customer's requirements are agreed to by all affected groups, the commitments between the engineering groups are agreed to by the affected groups, and that the engineering groups identify, track, and resolve intergroup issues. [Ref. 8:pp. 98-100]

• Peer Reviews - Such reviews aim to remove defects from the software products early and efficiently. An important corollary is to develop a better understanding of the software process and potential defects that can be prevented. It requires a methodical examination of work products by the producer's peers to identify defects and areas where changes are needed. Possible alternative ways of implementing peer reviews include Fagan-style inspections, structured walkthroughs, and active reviews. The goals are that peer review activities are planned and that defects in the software work products are identified and removed. [Ref. 8:pp. 101-103]

4. Level 4: Managed

At the managed level, the software development is well-defined and the organization sets quantitative quality goals for both products and processes. Both the software process and products are quantitatively understood and controlled. The principles of statistical process control are used to measure the software process and product quality for important activities across all projects. The results are stored in an organizational database, which is used to collect and analyze the data available from a project's defined processes. These measurements are used in evaluating a project's processes and products. Projects control their products and processes by narrowing the variation in performance

to within acceptable quantitative ranges. Sources of variations are discovered and corrected. [Ref. 7:p. 22]

SEI summarizes the software process capability of Level 4 organizations as being quantifiable and predictable because the process is measured and operates within acceptable measurable limits. This level of capability lets an organization predict trends in process and product quality. Because the process is both stable and measured, when some exceptional circumstance occurs the organization is able to identify and address their causes. When the measurements exceed acceptable ranges, managers take action to correct the situation. [Ref. 7:p. 22] There are two KPAs for this level:

• Quantitative Process Management - The KPA quantitatively controls the process performance of the software project. It involves establishing goals for process performance, measuring the performance of the project, analyzing these measurements, and making adjustments to maintain process performance within acceptable limits. Its goals are that the quantitative process management activities are planned, the process performance of the project's defined software process is controlled quantitatively, and the process capability of the organization's standard software process is known in quantitative terms. [Ref. 8:pp. 108-112]

• Software Quality Management - The goal is to develop a quantitative measurement of the quality of the project's software products and specify and achieve quality goals. It requires defining quality goals for the software products, establishing plans to achieve these goals, monitoring and adjusting software work products and development activities to meet quality goals to satisfy the needs of the customer. The goals for this KPA are

that the project's software quality management activities are planned; measurable goals for software product quality and their priorities are defined; and that actual progress toward achieving the quality goals for the software products is quantified and managed. [Ref. 8:pp. 113-114]

5. Level 5: Optimizing

At the optimization level, the entire organization is focused on continuous process improvement. Continuous process improvement is based on quantitative feedback. Project teams analyze defects to determine their cause, evaluate the process to prevent known types of defects from reoccurring, and disseminate lessons learned to other projects. Reducing waste happens at all maturity levels, but is the focus of Level 5. Waste is unacceptable, and organized efforts are directed at removing waste. Data on process effectiveness are used to perform cost benefit-analysis of new technologies and propose changes to the process. Innovations that exploit the best software engineering practices are identified and used throughout the organization. The software process is continuously improved in a controlled manner. [Ref. 7:p. 23]

SEI summarizes the software process capability of Level 5 organizations as "continuously improving." Because these organizations continuously strive to improve their process capability, they can expect to improve the quality of their products. Improvement occurs both by incremental advancements in the existing process and by innovations in technologies and methods. Technology and process improvements are planned and managed as part of the normal process of doing business. Level 5 is not the

final destination but the foundation for building an ever improving process capability. [Ref. 7:p. 23] The Level 5 KPAs are as follows:

• Defect Prevention - Its purpose is to identify the cause of defects and prevent them from reoccurring. This involves analyzing defects that were encountered in the past, and taking specific actions to prevent their reoccurrence in the future. Its goals are that defect prevention activities are planned, common causes of defects are sought out and identified, and common causes of defects are prioritized and systematically eliminated. [Ref. 8:pp. 118-120]

• Technology Change Management - The KPA identifies new technologies and incorporates them in the organization's standard software process in an orderly manner. New technologies include tools, methods, and processes. The goals for this KPA are that incorporation of technology changes is planned, new technologies are evaluated to determine their effect on quality and productivity, and appropriate new technologies are transferred into normal practice across the organization. [Ref. 8:pp. 121-123]

• Process Change Management - Its aim is to continuously improve the software process with the intent of improving software quality, increasing productivity, and decreasing the cycle time for product development. This involves defining process improvement goals, and systematically identifying, evaluating, and implementing improvements to the organization's standard software process and the project's defined software processes. The goals are that continuous process improvement is planned, participation in the organization's software process improvement activities is organization-

wide, and that the organization's standard software process and the projects defined software processes are improved continuously. [Ref. 8:pp. 124-125]

D. PLANNED IMPROVEMENTS

During the next few years, the current version CMM will continue to undergo extensive testing through the use of SPAs, SCEs, and process improvement programs. Feedback from these activities, Government, and the software industry will be used to improve the CMM. Version 1.1 is expected to be in use until at least 1996. The release of CMM version 2.0 is planned for the 1996-1998 time frame. This provides a balance between the need for stability and the goal of continuous improvement. [Ref. 8:p. 27]

Version 2.0 will incorporate suggested improvements by the user community. SEI is also working with the International Standards Organization (ISO) in an effort to build international standards into the CMM, SPA, SCE, and other process improvement activities. [Ref. 7:p. 27] While all levels of the model may be revised, the focus will be directed at Levels 4 and 5 [Ref. 8:p. 141]. SEI believes that the KPAs for Levels 2 and 3 have been almost completely defined. Since few organizations have been assessed to be at Levels 4 or 5, little is known about the characteristics of such an organization. The KPAs of Levels 4 and 5 will be refined as SEI works closely with organizations striving to understand and achieve these levels. The CMM may also expand its scope to address technology and human resource issues. [Ref. 9:p. 5.3]

E. SUMMARY

This chapter introduced the reader to the CMM. The SEI developed the CMM to assist software development organizations in increasing their process capability. The SEI is constantly improving the CMM based on feedback from the CMM user community.

The next chapter will provide an overview of the SCE, which is one of the CMM-based assessment tools created by the SEI.

III. THE SOFTWARE CAPABILITY EVALUATION

The SEI's Software Capability Evaluation (SCE) is a method for evaluating the software development process of an organization [Ref. 5:p. 15]. Its primary purpose is to support the acquisition of software by assessing an offeror's software process capability. A basic tenet of the SCE Method and CMM is that the more mature the contractor's software process capability is, the more likely it is that the contractor can meet cost, schedule, and performance targets.

The SCE provides a snapshot of a contractor's past process implementation, current process activities, and future process potential. Four to six Government personnel conduct a three day in-plant review at an offeror's facility. The results of the SCE are the strengths, weaknesses, and improvement activities measured against the CMM. [Ref. 10]

Use of the SCE method within DOD indicates that SCEs provide the following benefits for PMs:

- Adds Software Development Capability Realism to the Source Selection Process - The PM can compare SCE results with the evaluation of the contractor's proposal and software development plan (SDP) to determine whether the proposed approach is realistic in light of the offeror's current process capability.

• Increases Objectivity In Information Collected For an Acquisition - The SCE method helps ensure an objective review by putting a trained evaluation team on site to evaluate the offeror's activities and compare them against a public standard, the CMM. Each offeror is evaluated using the same criteria.

• Motivates the Contractor's Software Process Improvement Actions - By making SCE results a discriminator in source selection, contractors should be motivated to focus on software process improvement in order to retain or increase Government business. [Ref. 10:pg. 3-52 - 3-53]

A. DEVELOPMENT

At the request of the USAF, SEI and the MITRE Corporation conceived and developed the SCE Method in 1987. It was designed to help PMs determine the software process capability of a contractor at one organizational site (facility or location). [Ref. 9]

The original version of the SCE was first described in the 1987 SEI publication ***A Method for Assessing the Software Engineering Capability of Contractors***. Over several years, major changes to the original version have occurred based on feedback from SCE users. These changes include the elimination of maturity level scores, shifting from being a "question-based" to a "model-based" method, and public baselining of the SCE Method. [Ref. 5:p. 29]

B. CRITICISMS AND CHANGES

The SCE methodology and its use have not gone unchallenged. The result of the original SCE method provided to a Source Selection Evaluation Board (SSEB) was a

calculated maturity level for an organization's software development process. A contractor was scored as a "Level 1" to "Level 5" organization. This was based on a formula that used the number of verified "yes" answers to the SEI's maturity questionnaire by the contractor. [Ref. 5:p. 29]

There were problems with expressing the SCE results as a single maturity level score. The single maturity level score did not provide the SSEB with the detailed information on a contractor's software development process that it required to evaluate an offeror [Ref. 5:p. 29]. In addition, issues have been raised by the software industry on the statistical reliability of the formula used to calculate an organization's maturity level:

> By breaking the test into a multihurdle structure, the statistical reliability of the answer set is reduced in two ways. First, each hurdle or minitest will be less reliable simply because it has fewer questions. Second, the way the test is linked into a chain means that the uncertainty of the individual minitests must be multiplied by each other. Taken together, these effects result in a very rapid escalation of statistical uncertainty as the total number of hurdles is increased. [Ref. 11].

More recently, there has been a concern with Government organizations specifying that a minimum maturity level be achieved in order to be considered responsive to a solicitation [Ref. 12][Ref. 13]. To correct some of these problems, the SCE version 1.5 results are expressed as strengths, weaknesses, and improvement activities at the KPA level [Ref. 5:p. 29].

Another major change from the original SCE version was to shift the focus of the evaluation from the maturity questionnaire to the CMM. The original goal of the SCE was to validate a vendor's answers to the maturity questionnaire. There were two major

problems encountered when using the original SCE. First, not all KPAs were adequately covered by the questionnaire, and second, some questions were not based on KPAs at all. [Ref. 6:pp. 29-30]

To resolve these problems, the SEI has shifted the focus of the SCE version 1.5 from validating an offeror's response to the maturity questionnaire towards examining the KPAs of the maturity levels of the CMM [Ref. 6:p. 30]. In addition, a SEI representative stated in an interview with the researcher that the SEI has just published a revised questionnaire correcting these problems.

Another major criticism was that there was no public description of the SCE process. Detailed information about the SCE was only available through the training given to Government SCE teams. The secrecy of the method, especially considering the millions of dollars of potential Government contract work at stake, drew criticism from the software industry [Ref. 11:p. 28]. In July 1993, the SEI published ***CMU/SEI-93-TR-17 Software Capability Evaluation (SCE) Version 1.5 Method Description***. The SCE method description publication establishes a baseline. This will permit input into the future evolution of the SCE by organizations other than the SEI and SCE users.

C. SCE VERSION 1.5

The SCE version 1.5 consists of five major activity phases: Evaluation Start, General Preparation, Specific Preparation, Site Data Collection, and Findings. The method is summarized in Figure 1. A brief discussion of each phase follows.

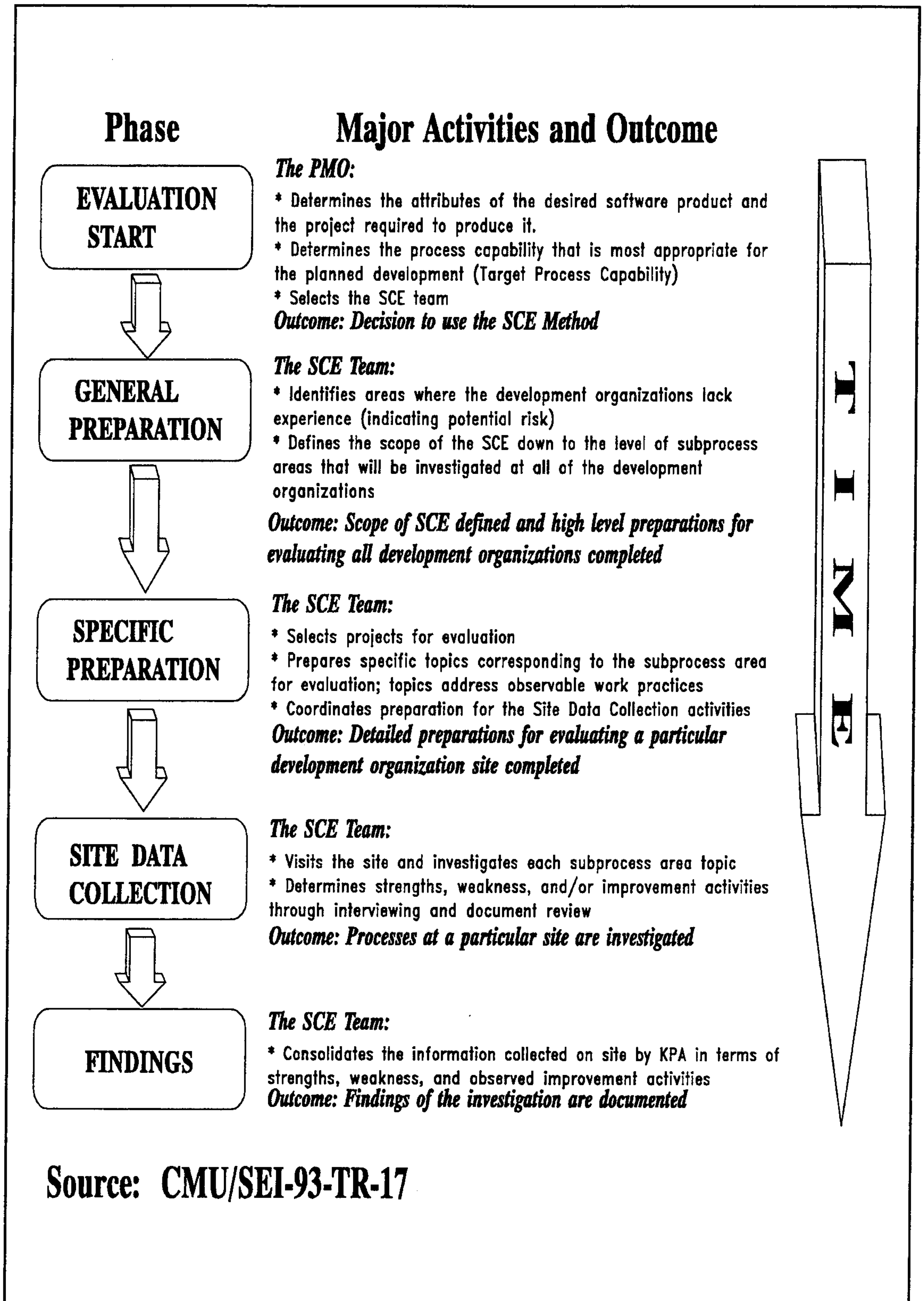


Figure 1 SCE Phases and Major Activities

1. Phase 1: Evaluation Start

In this phase, the Program Management Office (PMO) decides to use the SCE method and begins the required planning and preparation. The purpose of this phase is to determine the role of the SCE, define the attributes of the desired software product, determine the process capability required for the acquisition, and select the SCE team. This phase is performed by the PMO. Subsequent phases are conducted by the SCE team. [Ref. 5:pp. 20-21]

The first step is to decide whether to use the SCE. The SCE method was not developed to be used for all software acquisitions. PMs should consider the costs and benefits of using the SCE, as well as the unique requirements of their program, when making this decision. For small acquisitions, the costs may exceed the expected benefits of conducting an SCE. Figure 2 provides an application guide to assist the PM in deciding whether to use the SCE. [Ref. 10:pp. 3-45 - 3-46]

For all developments that involve significant Mission Critical Computer Resources (MCCR) however, the SEI strongly recommends using the SCE for source selection. The success of such systems is heavily dependent on its embedded software. The process capability of prospective vendors must be evaluated to select the contractor with the highest probability of success. [Ref. 10:p. 3-47]

Once the decision is made to use the SCE, the PM begins planning the activities required to incorporate the SCE into the source selection process. Details to be addressed include: determining the weight of the results of the SCE; including the intention to use the SCE in the Commerce Business Daily (CBD) announcement, Acquisition Strategy,

Criteria / Decision	Critical Software	Dollar Value	Management Control	Software Precedence	Lifecyle Phase	Schedule Length	Software Size
Definitely use SCE	DOD Major Program MCCR System	Software > $5M	High Priority	Unprecedented System	EMD Phase	Upgrades, major modifications, or follow-ons expected	> 100 KSLOC
Strongly Consider Using SCE		Software > 30% Cost		Need defined, any software CSCIs unprecedented	DEM/VAL Concept Exploration	Development > 3 years System life > 10 years	> 50 KSLOC
Consider Using SCE	Non-MCCR system	Total EMD program > $10M		Precedented system	Operational readiness support	Program length >= 5 years	> 25 KSLOC
SCE Use Likely Not Appropriate		Software < $5M < 30% of total costs Total EMD < $10M	Low Priority		Production/deployment		< 25 KSLOC

Source: CMU/SEI-93-TR-18

Figure 2 SCE Usage Decision Making Criteria

Request For Proposal (RFP), and other important acquisition documents; scheduling of SCE activities; and identifying and allocating resources in terms of dollars, time, and personnel to support the evaluation. [Ref. 5:p. 21] The SEI has provided examples of the recommended wording to use for SCE input to the CBD and RFP [Ref. 10:pp. 5-87 - 5-111]. The SEI also provides a sample schedule of SCE activities [Ref. 10:pp. 4-64, 4-67].

To conduct a SCE, the PMO must first define the software development project by creating a target product profile. The target product profile represents the "customer's view" of the product to be built [Ref. 5:p. 150]. The target product profile defines the proposed software project in terms of major and minor attributes. Major attributes include the application domain, product type, size, type of work, use of subcontractors, and previous vendor experience in developing this product type. Minor attributes, for which estimates are made, include the programming language required, target hardware configuration, applicable developmental standards, the customer, host development system, and configuration management tools. [Ref. 5:pp. 171-174]

Once the target product profile has been created, the next step is to determine the target process capability - the process capability that is most appropriate to the planned development [Ref. 5:p. 17]. Senior software engineers within or assisting the PMO determine which KPAs are required to successfully develop the proposed software system. These KPAs are matched to the CMM to determine the highest maturity level required by a prospective vendor. This maturity level becomes the target process capability. [Ref. 5:pp. 43-45]

Based on the CMM principle that higher levels of maturity cannot occur without first satisfying all the KPAs of the lower levels, all KPAs from Maturity Level 2 to the target process capability should be included in the SCE, whether they were determined to be important to the software project or not. At a minimum, the KPAs of Maturity Level 2 (Repeatable) should be evaluated. [Ref. 6:p. 45]

During this phase, the PMO requests important information from the vendors that will be used in later phases. A proposed project profile depicts the "vendor's view" of the proposed system. It is similar in form to the target product profile. Six to eight project profiles (also in the form of the target product profile) of ongoing or completed projects are submitted by the vendor as candidates for evaluation during the site visit. These projects should be similar to the proposed project profile submitted by the vendor. Organization charts are requested for the entire organization and for the development teams of the candidate projects. Answers to the questionnaire by the organization as a whole and the project teams are also requested at this time. The request for information is usually made in the Request for Proposal (RFP). [Ref. 5:p. 53]

The last step in this phase is selecting the team members. A SCE team consists of four to six personnel trained in the use of the SCE. The team should average a minimum of seven years of software development or management. Experience in software acquisition is also helpful. At least two members of the team should have previous experience in conducting SCEs. No more than one member should have less than two years of professional software experience. Collectively the team must have knowledge and experience with the following:

• The application domain and product type.

• The management processes required to create an effective environment for the engineering and development of a software product.

• The major phases that engineering and development of a software product must go through.

• The support processes and management environment required to reduce or eliminate unnecessary rework within the engineering and development of a software product.

• The relationship between technology (in the form of methods and tools) and support processes. [Ref. 5:pp. 46-47]

2. Phase 2: General Preparation

In this phase, the SCE team completes the high-level preparations for evaluating all offerors. In the first phase, the scope of the SCE was defined to the KPA level. The purpose of this phase is to narrow the scope to critical subprocess areas of all the KPAs within the target process capability. [Ref. 5:p. 49]

To determine the critical subprocess areas, the SCE team compares the proposed project profile for an organization with its project profiles of the projects submitted as candidates for evaluation. The team looks for mismatched attributes. A mismatched attribute exists if an organization's project profiles do not match any of the proposed project profiles. The target project profile and the proposed project profile are also compared. Any significant differences in major attributes can also be treated as a mismatch. [Ref. 5:pp. 52-55] The SEI has developed tables that identify subprocess areas

that should be evaluated if mismatches in major attributes occur. The SEI has also identified subprocess areas that should be used for all SCEs. [Ref. 6:pp. 175-183] These subprocess areas identified for evaluation are combined into one list. The list is tailored by eliminating subprocess areas that are not part of targeted KPAs, and including at least one subprocess area for each target KPA. Additional subprocess areas are added within the targeted KPAs based on the judgment of the SCE team. The tailored list defines the critical subprocess areas that are evaluated during the SCE for each offeror. [Ref. 5:pp. 56-59]

3. Phase 3: Specific Preparation

In this phase, the SCE team conducts detailed preparations for conducting a three-day site visit for each offeror. During the general preparation phase, planning focuses on issues that affect all the site visits. In this phase, the SCE team focuses on preparing for each individual site visit. The high-level preparations accomplished during the general preparation phase are refined and tailored for each offeror. The purpose of this phase is to prepare the SCE team for a specific site visit. [Ref. 5:p. 24-25,62]

Site visits are only conducted on those offerors within the competitive range [Ref. 10:p. 4-66]. A minimum maturity level should never be used as a screening factor in determining contractors within the competitive range [Ref. 10:p. 5-89].

The first step in this phase is to select the projects that will be evaluated for each offeror. The SCE team's goal is to evaluate risk in the processes that the offeror is planning to use on the project. It does this by comparing the project profiles and proposed project profile submitted by an offeror. It then selects three to four projects that

closely resembles the proposed project profile in terms of the major attributes that were described earlier in this thesis. By examining the processes used in these projects, the team hopes to gain a clear understanding of the processes that will probably be used on the subject acquisition. [Ref. 5:pp. 64-66]

Once the projects are selected, the SCE team requests documentation for review. This typically includes policies, standards, procedures, and directives relating to software development both at the organizational level and from the development teams for the projects that have been selected for evaluation. [Ref. 5:p. 65]

The SCE team also creates a key issue worksheet for each offeror. All critical subprocess areas and mismatches identified during the general preparation phase are listed on the key issue worksheet. The team then takes the offeror's answers to the questionnaire and looks for inconsistencies and anomalies. An inconsistency is a contradictory response from the same project to two or more questions that relate to the same subprocess area. An anomaly is a contradictory response to the same question by two or more projects. Inconsistencies and anomalies are also recorded on the key issue worksheet. [Ref. 5:pp. 66-69]

Using the key issue worksheet, the SCE team develops a list of topics to be investigated for each offeror. Topics are generated on subprocess areas that have mismatches, inconsistencies, or anomalies. Topics address the policies; roles and responsibilities; non-human resources; procedures and standards; training; adherence to policies, standards and procedures; and improvement activities concerning these subprocess areas. [Ref. 5:pp. 69-73]

Due to time constraints, all topics cannot be investigated. The SCE team must use its experience to balance the need for adequate coverage of critical subprocess areas against the limited time allotted for the site visit. It must limit the topics to those that will yield the most critical information.

Once the list of topics has been finalized for each offeror, the SCE team begins planning the procedures for conducting the interviews. This is done in several steps. First, the team decides how much time will be allocated for each topic for the site visit. Next, the SCE team selects who within the offeror's organization should be interviewed. Selection is done by position, not by name. More than one person can be interviewed for a topic and a person can be asked about several topics. Questions to be asked are finalized and documented. Finally, the team coordinates the interview schedule with the offeror to ensure that the interviewees are available at the designated times. It also coordinates the use of adequate facilities and administrative support at the vendor's plant during the site visit. [Ref. 5:pp. 73-76]

In the last step of this phase, the SCE team develops an initial briefing that will be given to all vendors. The agenda should include introducing the team members, describing the major on-site activities, discussing how the interviews will be conducted, and how the findings will be presented to the offeror. The initial meeting is recommended to last no more than 60 to 90 minutes. [Ref. 5:pp. 76-77]

4. Phase 4: Site Data Collection (Site Visit)

The Site Data Collection phase is the most important phase of the SCE method. It is during this phase that the SCE team evaluates the processes at a vendor's site.

The purpose of this phase is to investigate the topics associated with each critical subprocess area in sufficient depth to determine strengths, weaknesses, and improvement activities for the corresponding subprocess area. [Ref. 5:p. 79]

After conducting the initial briefing developed in the previous phase, the team begins collecting data. Site data collection has two basic components: investigation and decision making about the information collected. The team uses two complimentary means to investigate a topic: document reviews and interviewing. [Ref. 5:p. 79]

The SCE method assumes that a process not documented is usually not followed. Documents define and standardize processes, indicate commitment to use the process, provide an audit trail of processes used, and collect data about process performance. The documents requested earlier are reviewed for information. Additional documents may be requested to provide more information, clarify issues, or show evidence of use. The time required for documentation review varies at each site. [Ref. 5:pp. 26, 105-110] Examples of documents include organization charts, corporate policy/procedures on software management, project policy/procedures on software management, software development plans, software configuration management plans, software quality assurance plans, training plans, current metrics, and minutes of meetings [Ref. 11:p. 5-96].

Interviews provide insight into how the processes are implemented in practice, and show the extent to which the processes are understood by the developmental staff. The SCE method assumes that if a process is not understood by the people implementing it, it usually is not followed. [Ref. 5:p. 27] The number of people interviewed during a site

visit can be as few as ten or as many as 30 depending on the agenda developed by the SCE team during the specific preparation phase [Ref. 14].

Interviews are conducted involving one member of the development organization and the entire SCE team. The SEI believes that an employee will speak more freely without his supervisor and peers present. Also, data collection is more effective with the entire team conducting the interview. Because the entire SCE team is involved, the interview environment may be intimidating. Steps must be taken to put the interviewee at ease. [Ref. 5:pp. 111-113]

There are two types of interviews used during the site visit, exploratory and consolidation. The first round of interviews conducted by the SCE team are called exploratory interviews. These interviews are conducted in accordance with the published interview schedule given to the vendor. Exploratory interviews provide insight into how process areas are implemented and identify the documents where these processes are found. The length of an exploratory interview depends on the number of topics the interviewee is asked to address and the depth of the information required by the SCE team. [Ref. 5:pp. 85-87]

After the exploratory interviews are concluded, the SCE team conducts consolidation interviews to corroborate and clarify information that confirms or negates findings. During consolidation interviews, the SCE team questions individuals who may be able to provide additional objective evidence required to finalize the team's findings. This may include individuals questioned during the exploratory interviews. Consolidation

interviews usually focus on one or two questions aimed at resolving open issues. They are therefore generally shorter in duration than exploratory interviews. [Ref. 5:p. 93]

After each interview and document review, the team conducts a caucus. The purpose of the caucus is to analyze, share, and consolidate the considerable amount of data gathered so far. Team members share their analysis with each other to form a group consensus on what the data indicate and whether enough information has been gathered to make conclusions on a topic. If more information is required, the SCE team decides what is needed and which method to use, document review or interview, to get it. [Ref. 5:pp. 88-89]

A visual summary of the data collection process is depicted in Figure 3.

5. Phase 5: Findings

This is the final phase of the SCE in which the team documents the results of its investigation. The purpose of this phase is to consolidate the decisions made during the site data collection phase into findings at the KPA level. Decisions made by the SCE team during the caucuses address specific topics. These topics cover the critical subprocess areas, which in turn, are part of the targeted KPAs. The team takes the information it has collected and extrapolates the results to the KPA level in the form of strengths, weaknesses, and improvement activities. [Ref. 5:pp. 99-101]

The actual findings are normally completed at the vendor's facility, although the final report on the findings may be done later. If permitted by the contracting officer, the SCE team briefs the offeror on the results before leaving. A final findings report is tailored and provided to each offeror. It includes information common to all site visits

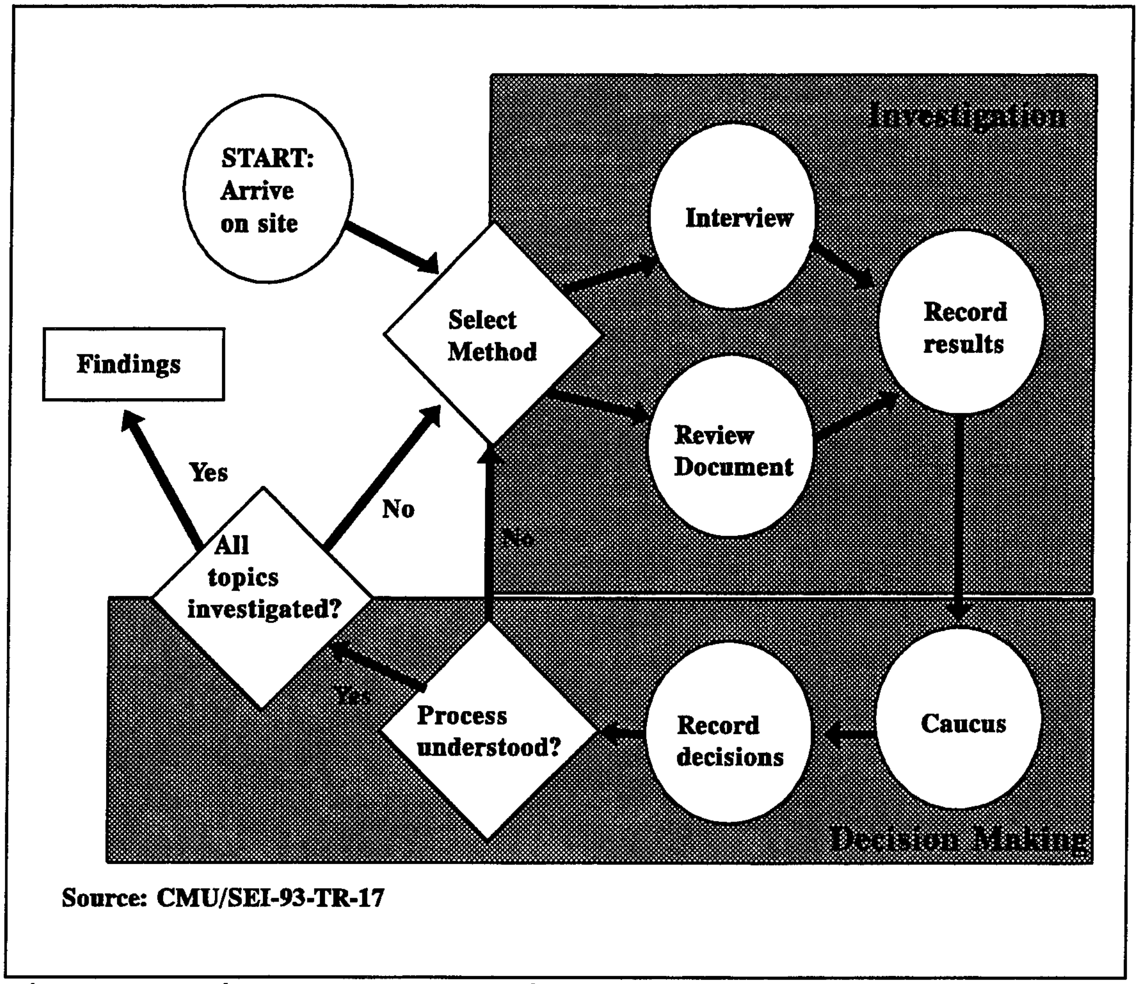


Figure 3 Site Data Collection Activities

such as the target product profile, target process capability, and critical subprocess areas. Information submitted by the offeror, including its proposed project profile, project profiles, organization charts, and other documentation, are also included. All worksheets, evidence used in forming the finding, and the actual findings are included in this report. The offeror is encouraged to incorporate these finding into its process improvement activities. Once the final report on the findings are published, and submitted to the SSEB, the SCE is completed. [Ref. 5:pp. 101-102]

The SEI makes a clear distinction that SSEB and Source Selection Advisory Council (SSAC) activities that incorporate the results of the SCE as part of the source selection process are separate from the SCE method [Ref. 5:p. 97]. SEI recommends that the SCE team leader be a member or advisor to the SSEB to work with the SSEB members in applying the evaluation standards to the SCE findings of each offeror [Ref. 10:p. 5-110]. This duty is performed outside the SCE, however.

D. SUMMARY

This chapter provided the reader with an overview of the SCE method. The benefits of the SCE include gaining insight into an offeror's software process capability and promoting process improvement within the software industry. In the next two chapters, the reader will be introduced to the SDCE and its model.

IV. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION MODEL

The SDCE method has two major components, a model that structures the SDCE criteria and questions, and a set of activities used in applying the model to a specific source selection [Ref. 2:p. 12]. This chapter begins with a description of the SDCE method, its purpose, and expected benefits and limitations. The remainder of the chapter provides an overview of the SDCE model. A more detailed description of the model criteria and questions are found in Chapter VI of ***AFMC Pamphlet 800-61 Acquisition Management Software Development Capability Evaluation Volume 1***. The following chapter will discuss the SDCE activities for applying the SDCE model.

A. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION METHOD

From several interviews with individuals involved with the development of the SDCE, the author learned that in August 1992, AFMC created a process action team (PAT). Its original purpose was to evaluate the SCE and SDCCR with the goal of adopting one method for use in AFMC source selections. The PAT's findings concluded that neither method was completely suitable. The PAT was then tasked to develop a software capability evaluation for AFMC. The result was the SDCE.

An ASC/AFMC representative stated in an interview that the SDCE method has been approved for pilot use on the Tri-Service Standoff Missile (TSSM) and three other

programs from the Space Systems Division. SDCE preparations are ongoing to support source selection activities for these programs which are scheduled for fiscal year 95. Pending the results of the SDCE's trial use, it is AFMC's goal to adopt the SDCE as its only software capability evaluation.

SDCE is a method used during source selection to evaluate an offeror's software engineering and management capabilities. It also evaluates an offeror's systems engineering capabilities which directly impacts software development. [Ref. 3:p. 1]

The primary purpose of the SDCE is to increase the probability of selecting an offeror capable of successfully developing software that meets all requirements within cost and schedule constraints. Its secondary purpose is to gain contractual commitment to implement the methods, tools, practices, and procedures necessary for success. [Ref. 3:p. 2]

The SDCE was designed to be used on all programs where software development is vital to the system. It is primarily applicable for use in source selection for the Engineering and Manufacturing Development Phase (EMD) contract. But it may also be used for the Demonstration and Validation Phase (DEM/VAL) and for major system modifications and/or upgrades contracts. It is intended to be applied to the prime contractor and its associate contractors and subcontractors involved in the planning and development of software. [Ref. 3:p. 4]

Although the SDCE method assists the SSA in selecting a capable contractor, the SDCE has limitations as well as benefits. The benefits include:

• The offeror is required to provide a comprehensive description of the software development capabilities in terms of engineering and management processes, methods, tools, and resources it proposes to use on the project.

• The SDCE method reviews the systems engineering and other development disciplines and processes that are directly related to software development.

• The SDCE method seeks to gain contractor commitment to follow well-defined processes described in the software development plan and are tied directly to the systems engineering master schedule.

• It promotes team building between the Government and the contractor by developing a mutual understanding of the offeror's capability and processes during the site visit.

• It also reduces program risk by focusing on the weaknesses in a contractor's software capability and process early in the development of the program.

The limitations of the SDCE method include:

• It does not develop program requirements found in the statement of work, specifications, and the contract data requirements list.

• Although the SDCE reviews an offeror's methods for estimating and generating software development schedules, it does not ensure an achievable software development schedule will be established and followed.

• It does not specify a software design solution to the program requirements.

• It cannot ensure that the resources that were evaluated during the SDCE will in fact be applied by the contractor to the program after contract award. [Ref. 3:pp. 4-5]

B. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION MODEL

The SDCE model contains critical capabilities that have historically been the high-risk areas in the development of software-intensive systems. The model organizes these critical capabilities into a three-level structure to facilitate the use of the SDCE method during source selection. The model is important because it contains the criteria and questions used by the SDCE team in evaluating an offeror's software development capability.

C. MODEL DEVELOPMENT

The SDCE model was developed from many sources. It is primarily based on elements of the CMM, SPA, SCE, and the SDCCR. These models and methods have been used extensively in the USAF for acquisitions and process improvements. They have all been subjected to extensive reviews. While they formed the baseline for the SDCE model during development, the model corrects some of the shortfalls that were previously identified in feedback from the software community. For example, the CMM does not address systems engineering issues in software development, or human resources. The SDCCR did not evaluate process improvement efforts, defect prevention, metrics, and technology assessment and transition. [Ref. 3:p. 12]

As mentioned previously, AFMC created a PAT to develop the SDCE method. The PAT consisted of personnel from AFMC and the SEI to address Government requirements, and industry representatives to address their needs and provide industry input into the development of the SDCE method. The PAT was composed of two teams.

One team developed the SDCE model, while the other team created the SDCE activities for applying the model. [Ref. 15]

D. MODEL STRUCTURE

The SDCE model is composed of three parts: the model structure, a set of model criteria, and questions. The model structure is a hierarchical decomposition of the capabilities deemed critical in the source selection process for software intensive systems. The structure facilitates the consolidation of SDCE findings in terms of strengths, weaknesses, and risk to the model structure level required by the source selection standards. The model structure has three levels:

• Critical Capability (CC) - This is the lowest level of the model structure. A CC is a set of model criteria that, when evaluated together, provide the basis for identifying strengths, weaknesses, and risks. [Ref. 3:p. 260]

• Critical Capability Area (CCA) - A CCA is a set of CCs that constitute an integrated development capability. The CCA facilitates the consolidation of strengths, weaknesses, and risks that were identified at the CC level. [Ref. 3:p. 260] • Functional Areas (FAs) - This is the highest level of the model structure. An FA is a set of related CCAs functionally grouped into major development capability areas. FAs are used to consolidate the strengths, weaknesses, and risks identified at the CCA level. [Ref. 3:p. 260]

The remaining components of the SDCE model are the model criteria and the questions. Model criteria are the defined standards for evaluating an offeror's capability

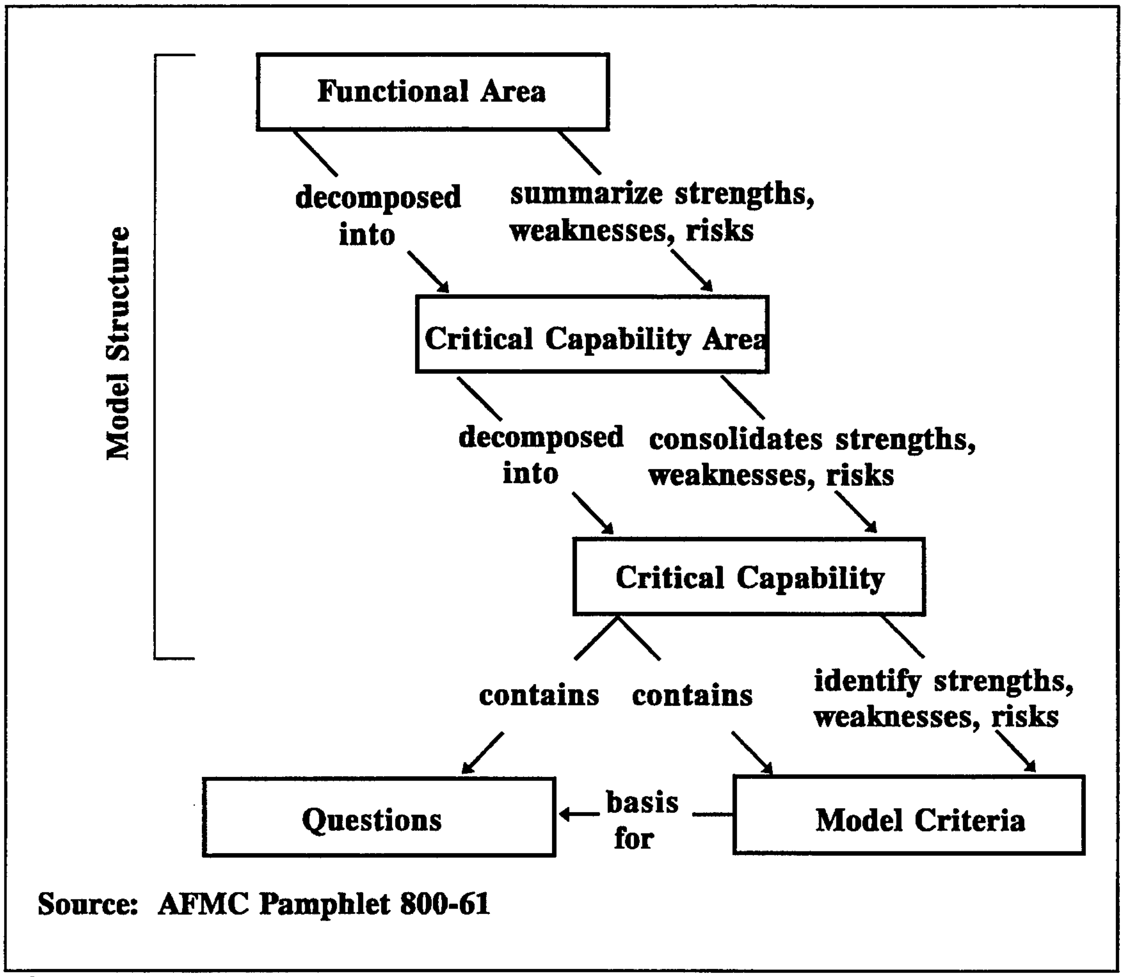


Figure 4 The SDCE Model Structure

and capacity in a standard and repeatable way. Model criteria address processes, people, tools, and technology. The questions part of the model consists of questions that are used to elicit information for specific model criteria. The questions ask an offeror to describe how it meets the model criteria. [Ref. 3:p. 14] Figure 4 depicts the model structure. Figure 5 provides an example of the SDCE criteria and questions.

FA: Organizational Resources and Program Support
CCA: Technology Assessment and Transition
CC: Technology Monitoring and Assessment

C1 **Systematic efforts are made in the organization to identify and assess new technologies that might meet identified or anticipated needs. Q1**	**Q1** **How do you maintain awareness of commercially available technologies that might meet identified or anticipated needs? How do you maintain awareness of leading relevant technical work? What is you approach for gathering and reviewing documentation of experiences with using these technologies? C1**
C2 **Information on advanced technologies in use in the organization, which could benefit other programs is disseminated. Q2**	**Q2** **How do you maintain awareness of advanced technologies in use in the organization? What information on these technologies do you disseminate to benefit other programs? How do you disseminate this information? C2**
Source: AFMC Pamphlet 800-61	

Figure 5 Example of Model Criteria and Questions

E. SOFTWARE DEVELOPMENT CAPABILITY EVALUATION MODEL FUNCTIONAL AREA OVERVIEW

There are six interrelated FAs for the SDCE model. Three of the FAs are layered on top of each other. Software engineering is part of systems engineering. It, in turn, is part of program management. The remaining three FAs cut across the first three FAs, in various degrees. For example, human resources is a part of the organizational resources and program support FA, yet it affects program management, software engineering, and systems engineering FAs. This is illustrated in Figure 6. [Ref. 3:p. 15]

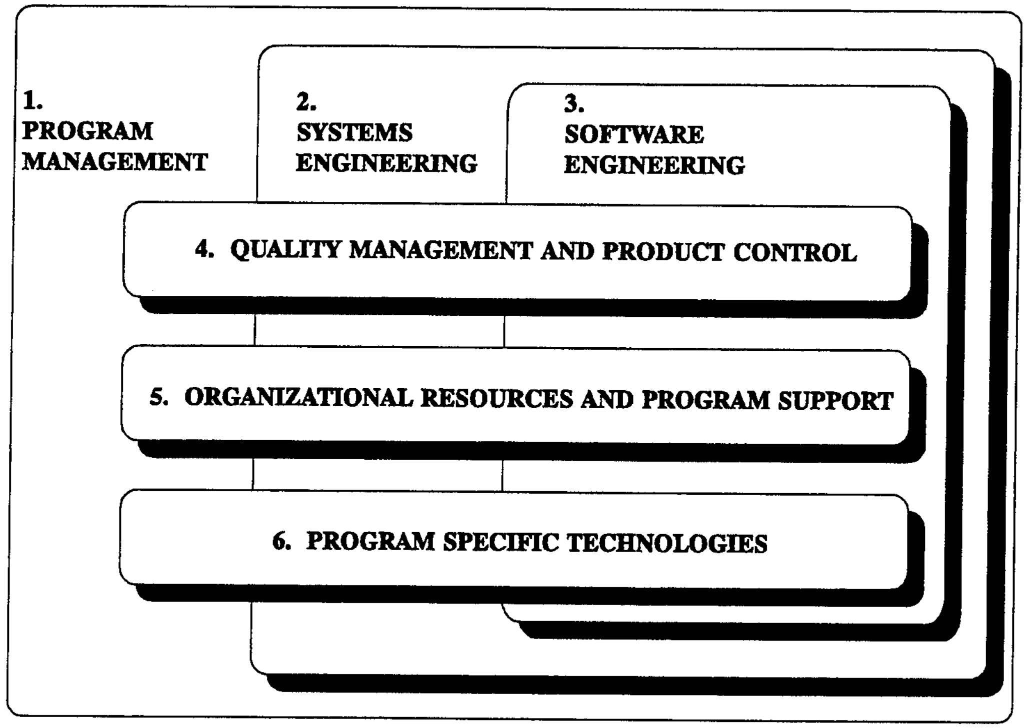
In the SDCE model, CCAs and CCs are assigned to only one FA even though they may be applicable to multiple FAs. CCAs and CCs are assigned to an FA based on a "best fit" determination based on the decisions of the SDCE PAT. [Ref. 3:p. 15]

The SDCE model is depicted in Figure 7. The remainder of this chapter provides an overview of the FAs and CCAs of the SDCE model.

1. Functional Area 1: Program Management

The program management FA focuses on program management capabilities required to successfully develop and manage software. Its purpose is to evaluate whether program level management processes and procedures are established, and if they support the software engineering processes and procedures being used. The program management FA should also accurately track the cost, schedule, and technical progress of the program. [Ref. 3:pp. 15-16] This FA contains the following CCAs:

• Management Authority, Responsibility, and Accountability - This CCA focuses on the organization structure and control processes. It evaluates the assignment of



Source: AFMC Pamphlet 800-61

Figure 6 SDCE Functional Areas

responsibilities, span of control, and the interrelationship between software engineering, program management, and systems engineering.

• Program Planning and Tracking - This CCA evaluates the processes used in program planning, developing the contract work breakdown structure (WBS), defining work packages, and defining the program schedule. The correlation between these planning and tracking processes is also evaluated.

• Subcontractor Management - This CCA evaluates an offeror's processes to control, maintain status, and report subcontractor development efforts. In particular, it looks at

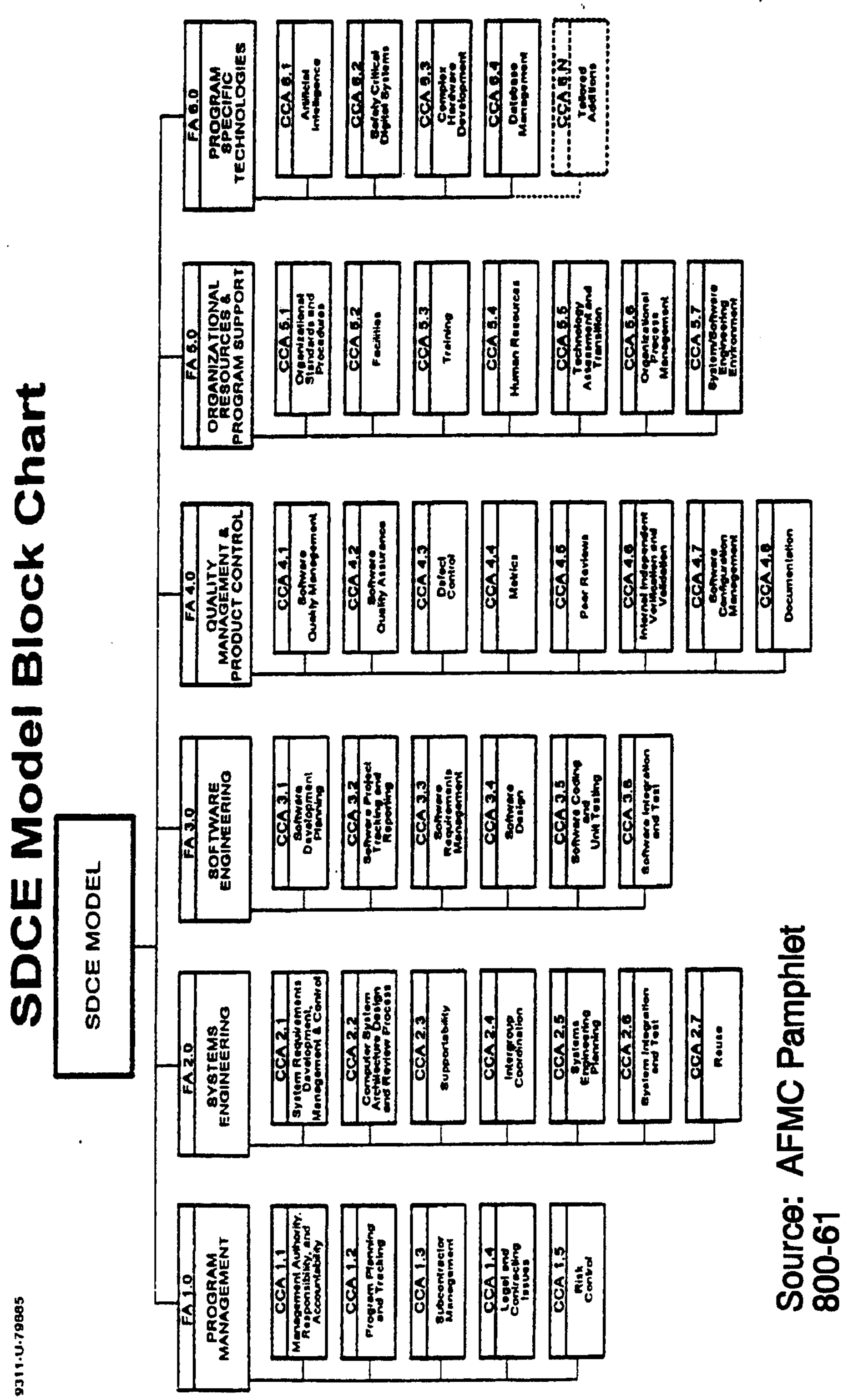


Figure 7 SDCE Model

the flowdown of development requirements through the Systems Engineering Management Plan (SEMP), Systems Engineering Master Schedule (SEMS), and Systems Engineering Detailed Schedule (SEDS) to the subcontractor. It also includes the review, test, integration, and software development planning of the subcontractor. The integration of subcontractor activities with the prime is also evaluated.

• Legal and Contracting Issues - This CCA evaluates the offeror's process for identifying proprietary and restricted rights software. It also looks at the procedures used to develop and support the software under restricted rights constraints.

• Risk Control - This CCA evaluates an offeror's process for identifying and managing program risks. [Ref. 3:pp. 26-28]

2. Functional Area 2: Systems Engineering

This FA focuses on those areas of systems engineering that have the greatest potential impact on the successful development of software. [Ref. 3:p. 16] It contains the following CCAs:

• System Requirements Development, Management, and Control - This CCA looks at the processes for developing and allocating system level requirements down to the software level. It also determines whether the resulting software requirements are adequate. The procedures for managing requirement changes, ensuring requirement traceability from the system to software level, and whether software issues are addressed during requirements development at the systems level are also addressed.

• Computer System Architecture Design and Review Process -This CCA addresses processes used to define the system level architecture design that includes hardware and

software components. It also evaluates the effectiveness of the system architecture design reviews and the procedures used to analyze the impact of changes to the architecture.

• Supportability - This CCA evaluates processes that addresses reliability and maintainability issues. These are of great concern to the support organization.

• Intergroup Coordination - This CCA looks at the processes that coordinate issues across different development groups. It also assesses the coordination among developer, customers, users, and testers.

• Systems Engineering Planning - This CCA evaluates the processes for defining systems engineering methods and the coordination with the software engineering methods to be used. It also looks at the adequacy of the SEMP, SEMS, and SEDS.

• System Integration and Test - This CCA assesses the process for integrating and planning. It also looks at the adequacy of the tools and facilities to support testing.

• Reuse - This CCA evaluates the processes used to exploit the advantages of reuse. It addresses the ability to reuse existing components, to develop common components, and to develop new components with increased reuse potential. [Ref. 3:pp. 28-30]

3. Functional Area 3: Software Engineering

This FA evaluates the capabilities that will be used to manage the engineering development of the software product. The capabilities include the ability to create an acceptable software development plan (SDP); estimate software size, cost, and schedule; define development methodologies; track and report against the SDP; and develop and control software requirements, design, coding, integration, and testing. [Ref. 3:pp. 16-18] This FA encompasses the following CCA:

• Software Development Planning - This CCA ensures that the resources required to meet all requirements are identified, budgeted, and allocated to the program.

• Software Project Tracking and Reporting - This CCA evaluates the process that keeps program and engineering management informed on the status of each software component and the program development as a whole. It also ensures that corrective actions are initiated and completed when necessary.

• Software Requirements Management - This CCA looks at the processes used to analyze, use, and maintain the software requirements after they have been baselined. (Recall that the initial development of the software requirements is covered in the systems engineering FA).

• Software Design - This CCA evaluates the procedures used to develop, document, and maintain the software design.

• Software Coding and Unit Testing - This CCA assesses the processes used to develop object code and to perform component or unit testing.

• Software Integration and Test - This CCA evaluates the processes used to integrate the software components then test them as a unit. [Ref. 3:pp. 30-32]

4. Functional Area 4: Quality Management and Product Control

This FA evaluates the processes that ensure and maintain software quality throughout the program's life cycle. Quality management entails defining, planning, implementing, and monitoring quality goals. Product control involves identifying the software configuration, systematically controlling changes to the configuration, developing documentation, and maintaining the integrity and traceability of the configuration

throughout the development life cycle. [Ref. 3:p. 18] This FA includes the following CCAs:

• Software Quality Management - This CCA examines management processes that support quality assurance functions. This includes defining quality goals, establishing plans to achieve these goals, and monitoring and adjusting the process to satisfy the needs of the customer.

• Software Quality Assurance (SQA) - The software quality assurance CCA looks at the vendor's quality assurance organization. This includes processes that ensure compliance with program standards and quality goals, reporting quality findings, and the elevation of unresolved quality problems to management levels above the project team.

• Defect Control - Defect control evaluates the processes that identify causes of defects and defect prevention. This includes thorough analysis of past defects and taking specific actions to prevent their reoccurrence in the future.

• Metrics - This CCA evaluates an offeror's processes to quantitatively assess the system and software development status and to consistently report metric results internally, to the prime and/or subcontractors, and to the customer.

• Peer Reviews - Peer Reviews are planned, methodical examinations of the software products by the producer's peers with the goal of early identification of defects and areas requiring change. This CCA examines the processes for conducting peer reviews.

• Internal Independent Verification and Validation (IIV&V) - This CCA looks at the processes for conducting IIV&V on critical software elements. It also ensures that the

development schedule can accommodate all the activities required for functional, performance, and documentation verification and validation.

• Software Configuration Management - This CCA evaluates the processes for establishing and maintaining the integrity of the software configuration. This includes processes for developing the software configuration, systematically controlling changes to the configuration, and maintaining the integrity and traceability of the configuration throughout the life cycle.

• Documentation - This CCA examines the processes used to develop and review the documentation required for the development. The documents include software requirement documents, software design documents, operator and maintenance manuals, test plans, and test procedures. [Ref. 3:pp. 32-35]

5. Functional Area 5: Organization Resources and Program Support

The purpose of this FA is to evaluate an organization's resources that are available and applied to the development program. It seeks to determine whether they are sufficient. [Ref. 3:p. 18] It includes the following CCAs:

• Organizational Standards and Procedures - This CCA looks at the areas that develops and maintains the organization's policies, standards, procedures, and other instruments that define the processes used. The organization should also update these items to reflect lessons learned.

• Facilities - This CCA ensures the required facilities to perform the system and software development functions are identified, developed, and allocated to the program.

• Training - Training develops the individual skills and knowledge required for personnel to perform their roles effectively and efficiently. This CCA assess a contractor's process for doing this.

• Human Resources - It is often heard that personnel are the greatest resource for the project [Ref. 2:p. 4]. This CCA ensures that human resources are available in the required quantities and skills. It effectively and efficiently manages the allocation of personnel to various developmental activities and also addresses retaining personnel for the life of the program.

• Technology Assessment and Transition - This CCA identifies and assesses new technologies that would be used on a project. It includes the tools, methods, and processes. When found to be beneficial to a project, the organization would transition them into use in an orderly manner.

• Organization Process Management - This CCA focuses on process improvement. It has the goals of improving quality, increasing productivity, and decreasing development time.

• System/Software Engineering Environment (S/SEE) - A S/SEE ensures the availability of integrated software development tools to support the different development and management functions. It should be consistent with the processes, methodologies, and languages used in development. [Ref. 3:pp. 35-37]

6. Functional Area 6: Program Specific Technologies

This FA addresses the technologies or application areas that are specific to a development program. Additional CCAs for unique technology or application areas

applicable to the program may be developed under this FA. [Ref. 3:pp. 18-19] This FA encompasses the following CCAs:

• Artificial Intelligence (AI) - This CCA evaluates an offeror's experience and expertise in applying AI tools and techniques to software development.

• Safety Critical Digital Systems - This CCA examines the offeror's capability and capacity to incorporate Safety Critical Digital Systems into the development.

• Complex Hardware Development - This CCA evaluates the offeror's processes and procedures for managing and developing complex custom integrated circuits.

• Database Management - This CCA evaluates the offeror's ability to develop large databases. [Ref. 3:pp. 37-39]

F. SUMMARY

ASC/AFMC and the SDCE PAT developed the SDCE as a method to evaluate the software development capability of an offeror during source selection evaluations. The SDCE model contains those critical capabilities that have historically been high-risk areas during software development. The activities that tailor and apply the SDCE model to a specific acquisition are discussed in the next chapter.

V. THE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION ACTIVITIES

The SDCE method is composed of 13 activities that encompass the six functional areas of the SDCE model. To distinguish them from the six functional areas, these activities are identified with letters A through M as depicted in Figure 8 [Ref. 3:p. 19]. Each of these 13 activities are further broken down into key tasks. This chapter will discuss the main features of each of the activities. The reader is directed to Chapter V of the AFMC publication for a more detailed discussion of the SDCE activities.

A. ACTIVITY A: DETERMINE APPLICABILITY

The decision to use the SDCE method must be made as early as possible in the acquisition cycle. This activity focuses on making the program office aware of the SDCE, familiarizing program office personnel with the SDCE method, assisting them in determining whether the SDCE is applicable and beneficial to their program, and obtaining a commitment to use the SDCE. [Ref. 3:p. 41]

Within each USAF Logistics and Product Center, SDCE center offices have been established to assist PMOs and SDCE teams in tailoring and applying the SDCE method. Local SDCE center offices and PMs have an equal responsibility to coordinate with each other as early as possible to begin SDCE planning. The local center SDCE office should be aware of all new programs that are initiated in order to advise PMs on the SDCE.

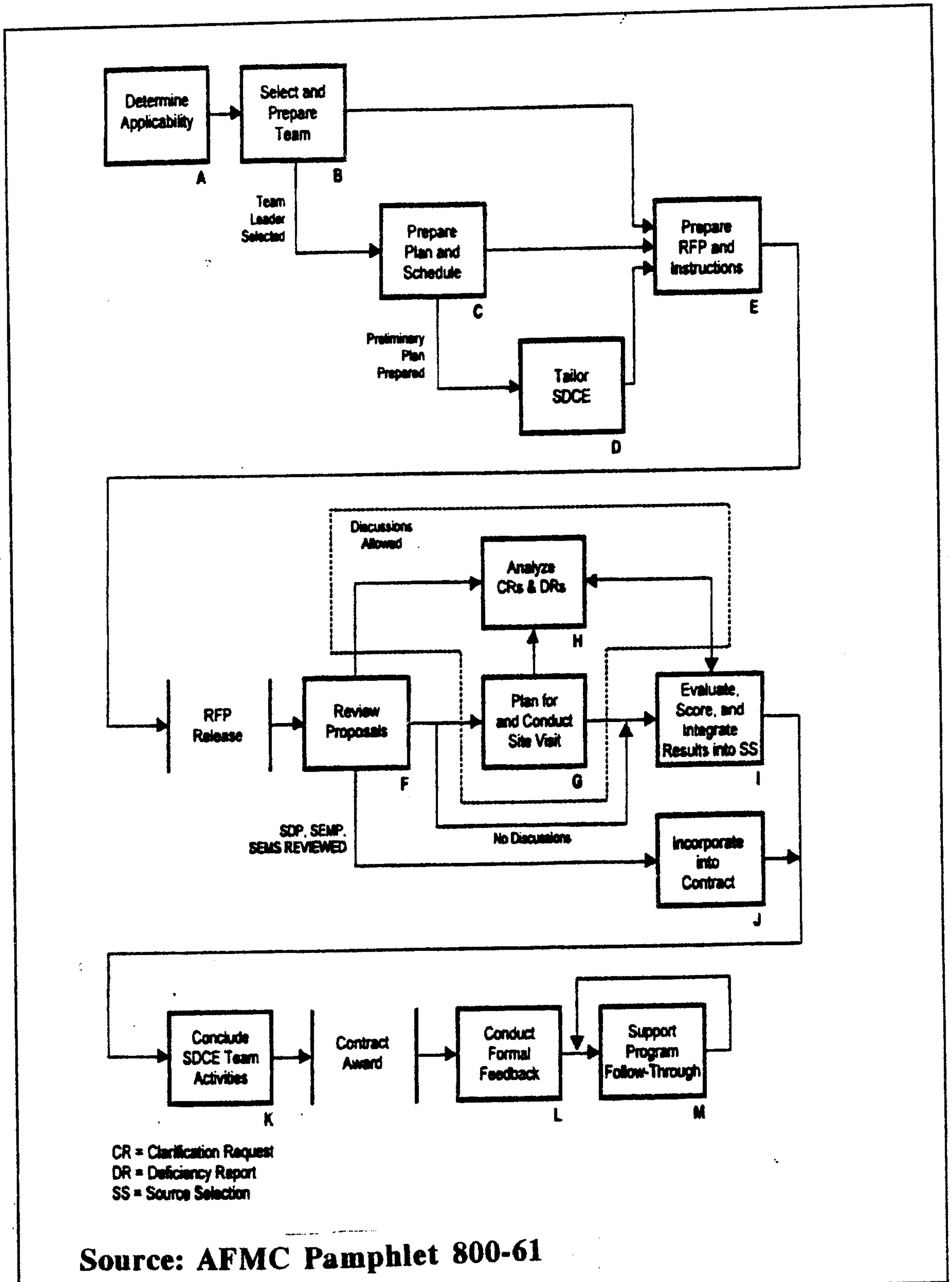


Figure 8 Overview of the SDCE Method

PMs should be aware of the existence of the SDCE method and its use, and should contact the local SDCE center office as soon as their program has been established. [Ref. 3:p. 41]

The first step for the PMO is to establish a working group to work with the local SDCE center office to determine the applicability of the SDCE to the subject project. This group should be composed of a PMO representative, the senior program/project engineer, the senior software engineer, and the procuring contracting officer. Ideally, these same individuals should be part of the SDCE team if the decision is made to use the method. [Ref. 3:p. 41]

The SDCE method is intended to reduce software development risks. An SDCE application guide is depicted in Figure 9. It identifies the factors that should be considered when determining the applicability of the SDCE to a program. Although not shown on this chart, the acquisition strategy must also be reviewed to determine the impact of the SDCE. [Ref. 3:pp. 43-45]

A PMO's resources are not unlimited. The cost of the SDCE must also be a factor in determining whether to use it. An estimate of the cost of conducting a SDCE is provided in Figure 10. [Ref. 3:pp. 45-46]

Once the PMO working group has determined the costs of conducting the SDCE and the benefits it expects from the evaluation, it briefs the PM on its findings and recommendations. If the PM decides to use a SDCE, key leaders involved with the acquisition, both outside and within the PMO, are briefed on the expected benefits and costs. Outside leaders include the Program Executive Officer (PEO), and directors from

The SDCE process should be applied during source selection on weapon systems entering the EMD phase when two or more of the following conditions, requirements, or characteristics exist. If only one exists, applying the SDCE may still be appropriate for particular acquisitions.

- The development is designated a major program.
- The system software is estimated to cost more that $25 million or require more than 100,000 SLOC.
- The program involves highly complex requirements and an associated complex software development process.
- A complex software to systems integration effort is expected.
- The software development is constrained by an aggressive program schedule or it is anticipated that the software development will be on the project's critical path.
- The program development involves safety critical software (e.g. human safety or nuclear surety factors).
- The program development includes unprecedented functional capabilities or new software technologies.
- It is anticipated that there may be bidders with uncertain software development and management capabilities, or unknown experience with the application domain.
- The program is software intensive.

The SDCE process should be considered for weapon system DEM/VAL source selections when either of the following conditions exist.

- Significant software developed during DEM/VAL is planned to be reused in the EMD phase.
- Major EMD software contractors are expected to be selected from the DEM/VAL phase contractors.

SOURCE: AFMC Pamphlet 800-61

Figure 9 Guidelines to Applying the SDCE

Activity	Effort (Person Days)
SDCE team preparation through final RFP release	20 - 80
SDCE team proposal analysis prior to site visit (per offeror)	18 - 24
SDCE team site visit (per offeror)	24 - 30
SDCE team evaluation (per offeror) after site visit	18 - 24

Source: AFMC Pamphlet 800-61

Figure 10 Estimated SDCE Costs

the supporting contracting, engineering, and quality assurance directorates. The goal is to gain their support, especially in committing the required resources. [Ref. 3:pp. 46-47]

B. ACTIVITY B: SELECT AND PREPARE TEAM

The selection of competent and experienced individuals as members of the SDCE team is crucial to the success of the evaluation. While a smaller team is desired for efficiency, a larger team offers technical depth, coverage, and experience. Such potentially conflicting requirements must be adequately balanced. [Ref. 3:p. 19]

The first step in organizing a team is to select the team leader. The team leader should have a minimum of 15 years of weapon systems acquisition and possess considerable knowledge in systems and software engineering. This person must also have the skills to lead small groups, and to convincingly communicate the results of the evaluation.

It is recommended that the team leader come from the PMO. This ensures the PM has visibility and input throughout the evaluation. It also provides a means to insure that the knowledge, experience, and shortfalls of the contractor's software development process discovered during the evaluation are assimilated into the PMO and considered during the project development. [Ref. 3:pp. 48-49]

The evaluation team is composed of two parts, a core team and a support team. The core team consists of the team leader and two or three additional senior members who are experienced with software engineering, project management, and logistics. Its role is to provide technical expertise to evaluate the offeror's software development capability. After the preliminary source selection planning has been completed and specific source selection parameters and conditions have been determined, the team leader and PMO select individuals for the SDCE team. Selections are based on the basis of the qualification criteria stated below and any unique requirements for the program. [Ref. 3:pp. 48-51]

The combined experience of the team should cover the following areas: systems engineering, software engineering, program management, and logistics engineering. Team members should also be knowledgeable in all the CCAs and CCs that will be used in the evaluation. [Ref. 3:pp. 49]

Core team members are normally members of the SSEB. They are expected to participate fully in the evaluation of the proposal. Recommended core team qualifications are listed in Figure 11. [Ref. 3:p. 49]

Core Team Qualification Criteria

The **Senior Systems Engineer** should have experience in the systems engineering process. This includes experience in transformation of validated customer needs and requirements into a set of system products and designs (MIL-STD-499B), requirements definition and specification, SEMP and SEMS, and systems engineering requirements in the RFP.

The **Senior Software Engineer** should have an extensive background and understanding of the software development process. This includes requirements analysis, design, code and test, integration, the software engineering process and procedures, project planning and estimation, and software engineering requirements used in the RFP.

The **Senior Project Manager** should have experience and understanding of the fundamental concepts and principles of project planning, process models, project scheduling, and milestones. The project manager should also have experience in project organization and management issues, development team organization, project costing, and management requirements for the RFP.

The **Senior Logistics Engineer** should have knowledge of all activities related to the development and support of software systems. The logistics engineer should be familiar with the software product life cycle, software support environments, software documentation and training, and the logistics requirements for the RFP.

Source: AFMC Pamphlet 800-61

Figure 11 Recommended Core Team Qualification

The support team supplements the core team with specific skills, knowledge, and experience required for the evaluation. They are only called upon when their experience/expertise is needed. They may not stay for the duration of the evaluation. Support team members provide expertise in such disciplines as: software engineering, software management, subsystems engineering, contracting, quality assurance, configuration management, test, and financial management. [Ref. 3:p. 49]

Once the team is formed, the team leader coordinates the preparations for the evaluations. First, the team meets with the major divisions of the PMO to familiarize themselves with the subject system, the issues related to its acquisition, and any special considerations. A review is made of key acquisition documents such as the acquisition strategy, mission need statement, operational requirements document, and request for proposal. After the team has become knowledgeable on the project and the acquisition strategy, it receives training on the SDCE method. [Ref. 3:p. 52]

C. ACTIVITY C: PREPARE PLAN AND SCHEDULE

In the early stages of planning, it is likely that only the SDCE team leader will have been identified. The team leader should work closely with the PMO, the Source Selection Authority (SSA), and the SSEB to ensure that adequate time is reserved to conduct SDCE activities required during source selection. These include the allowance of sufficient time for the team to evaluate each proposal and to conduct site visits. Site visits are made between the time the competitive range is determined and best and final offers (BAFOs) are required. A proposed schedule is provided in the AFMC pamphlet. [Ref. 3:pp. 53-56]

The team leader also works with the SSEB to determine how the SDCE results will be integrated in the source selection process. The normal approach is to use SDCE results as a factor under the "Technical" area. The actual approach can vary based on the determination of the SSA. [Ref. 3:pp. 54-58] A key decision made by the SSA is whether to award the contract with or without discussions. In either case, the team leader plans for site visits. The impact of a "no discussion" decision on the SDCE is that less

time may be available to conduct site visits. Site visits may not be required if sufficient information is gathered from the offerors to make a decision, even if discussions are planned. [Ref. 3:p. 60]

Once team members are selected, detailed planning occurs. As more specifics on the source selection schedule become available, the core team develops a definitive SDCE activity schedule. It also develops evaluation standards for all CCs to be evaluated using the SDCE model. The SDCE activity schedule and CC evaluation standards are included in the source selection plan. [Ref. 3:p. 61]

D. ACTIVITY D: TAILOR SOFTWARE DEVELOPMENT CAPABILITY EVALUATION, SELECT CRITERIA AND QUESTIONS

The purpose of this activity is to define the scope of the SDCE. When tailoring the SDCE model to a specific acquisition, the SDCE team creates a subset of the SDCE model. This subset contains the CCs that are identified by the SDCE team as high risk areas to the project development.

Tailoring is done in several steps. First, the SDCE team develops a software profile for the acquisition. To do this, the software project is defined in terms of: estimated software size, estimated development time, and estimated development team size. The evaluation team also identifies special technical areas important to the software project. These include the software language, tools, methods, systems/software engineering environment, complex integrated circuit development, open systems architecture,

commercial off-the-shelf software, reuse requirements, complex interface requirements, security/safety requirements, and portability. [Ref. 3:pp. 67-68]

Once the proposed software has been defined, the SDCE team members use their knowledge and experience to identify those CCs that should be evaluated during the source selection. These high-value discriminators must be included in the RFP. The documentation required from the offeror to evaluate these CCs are requested in the RFP. A site visit is only used to clarify and substantiate the information received from the offeror. The team cannot evaluate CCs not mentioned in the RFP. [Ref. 3:pp. 69-72]

As mentioned in the previous chapter, each CC in the SDCE model has defined questions and criteria. When the SDCE team selects CCs for the SDCE evaluation, it also automatically selects the evaluation criteria and questions. In some cases a team may develop unprecedented CCs unique to the acquisition that are not contained in the SDCE model. For example, a SDCE team may want to evaluate a contractor's ability to incorporate an unprecedented specific software technology on the proposed program. This is a significant undertaking. The SDCE team would work with the local SDCE Office to create the criteria and questions for such new CCs. [Ref. 3:p. 74]

E. ACTIVITY E: PREPARE REQUEST FOR PROPOSAL AND INSTRUCTIONS

In this activity, the SDCE team takes steps to make the offerors aware of the Government's intent to use the SDCE during source selection. This is done through several means. The CBD announcement for the acquisition should include a paragraph

declaring that a SDCE would be used. This would also be stated in the RFP. The SDCE team would also provide information to potential offerors on the SDCE method and procedures during the bidder's briefing. [Ref. 3:p. 77]

The team develops the SDCE input to the RFP. An important part of this input is the Government's request for information from the offerors. This describe how each high value CC is implemented and what would suffice as evidence of its use. Documents that describe how a CC is implemented include organization charts, job descriptions, implementation manuals, and internal standards. Evidence of use is provided in the form of minutes of meetings, evaluation reports, and metrics. This information should come from three to four completed or ongoing projects that are similar to the software being developed. [Ref. 3:pp. 75-76]

In the RFP, each offeror is directed to provide in their offers answers to the SDCE model questions related to the high value CCs that will be evaluated. These questions are also included in the RFP. Answers can be references to other documents like the SEMP, SDP, or company policy and procedures. [Ref. 3:p. 75]

The team also prepares the specific SDCE input to the General Notice to Offerors (GNTO) section of the RFP. It explains how the site visit will be conducted, the tentative schedule for site visits, the facilities and resources required by the SDCE team, and the types of contractor personnel who may be interviewed. [Ref. 3:pp. 78-80]

Examples of the recommended wording for the SDCE input to the CBD announcement and RFP are provided in the AFMC pamphlet [Ref. 3:pp. 1-14, 1-18 - 1-21].

F. ACTIVITY F: REVIEW PROPOSALS

The SDCE team examines an offeror's information submitted with its proposal for completeness and consistency. Answers to the SDCE model questions in the RFP should fully describe how the CCs being evaluated are implemented. All processes and procedures should be documented, and evidence of their use should have been submitted. The SDCE team also checks whether the proposed software development approach is consistent in the proposal, SEMP, SEMS, and SDP. After reviewing the information, the SDCE team performs an initial assessment of strengths, weaknesses, and risks, using the method described in Activity I. [Ref. 3:pp. 82-84]

Once the initial assessment is made, clarification requests (CRs) and deficiency reports (DRs) are created. CRs are used to request additional information to clarify a process or procedure, or to show evidence of use. CRs are also used to document inconsistencies between the SEMS, SEMP, and SDP. When a capability or procedure is clearly deficient with respect to RFP requirements, it is documented in a DR. CRs and DRs are released to the offeror if discussions are permitted. If discussions are not allowed, CRs and DRs are used in the evaluation to document weaknesses in an offeror's proposal. [Ref. 3:pp. 86-87]

G. ACTIVITY G: PLAN FOR AND CONDUCT SITE VISIT

The SDCE team plans for a three to four day site visit. The site visit has three purposes. First, it provides a forum for the SDCE team and the offeror to discuss and develop a mutual understanding of the proposed software development process. It also

provides a means to verify a vendor's ability to implement its proposal. Finally, it gives the SDCE team the opportunity to encourage the offeror to incorporate the proposed processes into the SDP, SEMP, and SEMS. [Ref. 3:p.88]

To prepare for the site visit, the SDCE team reviews the offeror's proposal and any responses to CRs and DRs to identify areas that still need clarification. It then develops an agenda and topics of discussion for the site visit. These are sent to the contractor no later than two weeks before the site visit. The team also coordinates with the offeror to ensure that adequate facilities will be available to support the team. [Ref. 3:pp. 89-90]

Once on site, the team briefs the offeror on the SDCE method and its overall use in the source selection. The team then begins discussions on the agenda items. Discussions are confined exclusively to the offeror's proposal, performance history, and CRs and DRs. The evaluation team looks for completeness and adequacy in the offeror's response. It also looks for a strong or weak level of compliance with the evaluation criteria. All information from the offeror is carefully recorded by the SDCE team. [Ref. 3:pp. 92-93]

At the end of the discussion, the SDCE team reviews any additional documentation provided by the offeror. It then holds a meeting to review the information collected and to develop a team understanding of the offeror's processes and capability. Afterwards, the team prepares a feedback presentation to the offeror. This is the last event of the site visit. The purpose of this presentation is to communicate the team's understanding of the offeror's processes and capability. The offeror is given the opportunity to respond to this briefing. [Ref. 3:pp. 93]

H. ACTIVITY H: ANALYZE CLARIFICATION REQUESTS AND DEFICIENCY REPORTS

CRs and DRs are generated throughout the evaluation process. These may be released to the offeror with the permission of the SSA. The offeror's response to CRs and DRs must then be analyzed; when required, follow-up CRs or DRs are generated. [Ref. 3:pp. 96-97]

I. ACTIVITY I: EVALUATE, SCORE, AND INTEGRATE RESULTS INTO SOURCE SELECTION

In this activity, the SDCE team evaluates the information received by the offeror and scores the results. An offeror's capability is scored in two areas. The first is the degree to which the offeror's capabilities met the evaluation criteria. The second is the overall risk of the offeror's proposed development plan. [Ref. 3:pp. 100, 105]

Evaluations are conducted from bottom to top. The SDCE team begins at the CC level then consolidates the score to the FA level. Capabilities are scored as either strong, acceptable, or weak. Risks are scored as either high, moderate, or low. Both the capability and risk scores are accompanied by supporting narratives. The guidelines for doing this are depicted in Figure 12. [Ref. 3:pp. 114-116]

Once an offeror's capability and risk have been scored at the FA level, the SDCE team assigns a color code and risk rating to the offeror's proposal. The color code and risk rating are used by the SSA or Source Selection Advisory Council (SSAC) to compare offerors. The SDCE uses its knowledge and experience to develop these scores. The

CCA Capability Roll-up Guidelines

RATING	DESCRIPTION
Strong	The majority of the CCs are strong and there are no significant weak CCs
Acceptable	The majority of the CCs are acceptable and there are no significant weak CCs
Weak	There are significant weak CCs

FA Capability Roll-up Guidelines

RATING	DESCRIPTION
Strong	The majority of the CCAs are strong and there are no significant weak CCAs
Acceptable	The majority of the CCAs are acceptable and there are no significant weak CCAs
Weak	There are significant weak CCAs

CCA Risk Roll-up Guidelines

RATING	DESCRIPTION
Low	The majority of the CCs are low risk and there are no significant high risk CCs
Moderate	The majority of the CCs are moderate risk and there are no significant high risk CCs
High	There are significant high risk CCs

Source: AFMC Pamphlet 800-61

Figure 12 SDCE Roll-up Guidelines to the FA Level

source selection scoring guidelines are listed in Figure 13. [Ref. 3:pp. 119-120]

Source Selection Color Codes

COLOR	RATING	DEFINITION
Blue	Exceptional	Exceeds specified performance of capability in a beneficial way to the Government, and has no significant weaknesses.
Green	Acceptable	Meets evaluation standards, and any weaknesses are readily correctable.
Yellow	Marginal	Fails to meet evaluation standards, however any significant deficiencies are correctable.
Red	Unacceptable	Fails to meet minimum requirements of the RFP, and the deficiency is uncorrectable without a major revision of the proposal.

Proposal Risk Ratings

SYMBOL	RATING	DEFINITION
L	Low	Has little potential to cause disruption of schedule, increase in cost, or degradation of performance. Normal contractor effort and normal Government monitoring will probably be able to overcome difficulties.
M	Moderate	Can potentially cause some disruption of schedule, increase in cost, or degradation of performance. However, special contractor emphasis and close Government monitoring will probably be able to overcome difficulties.
H	High	Likely to cause significant serious disruption of schedule, increase in cost, or degradation of performance even with special contractor emphasis and close Government monitoring.

Source: AFMC Pamphlet 800-61

Figure 13 SDCE Source Selection Scoring Guidelines

J. ACTIVITY J: INCORPORATE INTO CONTRACT

The purpose of this activity is to gain contractual commitment for the development processes being proposed by the offeror. Responses to the CRs and DRs are not binding by themselves. They must be incorporated into key development documents, which include the SDP, SEMP, and SEMS to be enforceable. [Ref. 3:pp. 124-125]

K. ACTIVITY K: CONCLUDE SOFTWARE DEVELOPMENT CAPABILITY EVALUATION TEAM ACTIVITIES

At this point, the major activities of the SDCE team have been accomplished. The team members derive the SDCE metrics that will be used to improve the SDCE method. SDCE metrics include problems encountered during the evaluations, new CCs developed, strengths and weaknesses of the SDCE method, and recommended improvements. The team also properly disposes of SDCE data no longer needed. The final action of this activity is the disbanding the team. [Ref. 3:pp. 126-127]

L. ACTIVITY L: CONDUCT FORMAL FEEDBACK

Former SDCE team core members may participate in briefings to the winner and unsuccessful offerors. They answer questions concerning the SDCE evaluations within the guidelines established by the contracting officer. They also solicit feedback on the vendor's experience with the SDCE. This feedback will be used to improve the SDCE method. [Ref. 3:pp. 128-130]

M. ACTIVITY M: SUPPORT PROGRAM FOLLOW-THROUGH

This activity involves using the SDCE results for purposes other than source selection. This could include input to the risk management plan and to an offeror's improvement plan. It can also include using the SDCE results to monitor the progress of the contractor, especially in those areas that were identified as weak in the contractor's software development capability. [Ref. 3:pp. 131-132]

N. SUMMARY

The last four chapters provided the reader with a basic knowledge of SCE, SDCE, and their respective models. The next chapter will present the findings of two DOD studies on software capability evaluations.

VI. LITERATURE RESEARCH

The previous chapters provided the reader with an overview of the SCE and SDCE methods, along with their corresponding models. This chapter presents the findings of two DOD studies that compared the SCE with the SDCE's predecessor, the SDCCR.

A. AEROSPACE CORPORATION STUDY

In June 1992, the Aerospace Corporation researched and published the report ***Pre-Award Survey Method for Software Acquisition at Space Systems Division*** in response to a USAF Space Systems Division (SSD) tasking [Ref. 16]. Although unable to obtain an actual copy of the study, the researcher was able to obtain information on the study by interviewing one of its authors.

In the interview, one of the authors of the Aerospace Corporation report explained that Aerospace Corporation is under contract with the SSD to conduct source selection software capability evaluations. The SSD directed the Aerospace Corporation to conduct a study to determine which evaluation, the SCE or SDCCR, should be adopted to assess the software development capabilities of offerors during the source selection of SSD programs. The study team examined five to six software intensive satellite and radar programs and identified areas of risks to the software development project. The study team then determined which of the two methods, the SDCCR or SCE, was better at evaluating these areas.

The study concluded that, although the SDCCR had weaknesses, the initial recommendation was to use the SDCCR. The long term recommendation was to combine the best of both approaches and create a new evaluation method. [Ref. 16:p. 46].

In the same interview, the author cited the following reasons for the study's conclusions:

- "The SCE assumes that software is developed in a vacuum." The SCE had a narrow focus of software development. It did not address key issues, such as systems engineering, and program specific technologies and tools that have an impact on software development. The SDCCR covered these areas.

- While answers to the "yes/no" type questionnaire used by the SCE indicate whether a process exists, they do not provide enough information to determine the adequacy of the process. The SDCCR essay type questions did.

- In the SCE, a process was only judged to be adequate if it was already in existence and its procedures and proof of past use were well-documented. No consideration was given to new processes being proposed for the project. For the SDCCR, the SDCCR team uses its knowledge and experience to evaluate new processes being proposed by the contractor for the specific project. Some consideration may be given if warranted.

- The SDCCR did have weaknesses. The structure of the SDCCR model made it difficult to tailor the method to a specific acquisition. The SDCCR also did not address important areas covered in the CMM, such as defect prevention, metrics, and process

improvement. This is why the study recommended combining the strengths of the SDCCR and SCE to create a new evaluation method.

B. INSTITUTE FOR DEFENSE ANALYSIS (IDA) STUDY

In 1992, the Defense Advanced Research Projects Agency (DARPA) (now known as the Advanced Research Projects Agency (ARPA)) tasked the Institute for Defense Analysis (IDA) to analyze policy issues involving a wider DOD application of the SEI software process maturity model [Ref. 16:p. v]. The IDA study focused on implementation issues associated with mandating software process assessments and software capability evaluations DOD-wide, the availability and adequacy of data to determine the benefits and effectiveness of process maturity, and a comparison of the maturity model and methods with similar techniques. IDA drew on its experience in performing SCEs in support of what was then called the Strategic Defense Initiative (SDI). It also used the results of a survey of 123 organizations within the SPA/SCE community. [Ref. 16:pp. v, 80]

A summary of the highlights of the IDA study includes the following:

• Although the SCE had weaknesses, it helps to promote desired process improvement, and SCE interviews verify an organization's current software development process capability. [Ref. 16:pp. 11, 51].

• Definitive data do not exist to quantify the benefits and effectiveness of process maturity. Anecdotal evidence of organizations experiencing significant positive return on

investment (ROI) by increasing their process maturity levels supports the pursuit of process improvement based on the CMM [Ref. 16:p. 32].

• Sixty-eight percent of the contractors who had responded to the survey and had been subjected to more than one SCE, felt that SCEs were useful in evaluating a contractor's software process capability [Ref. 16:p. 80].

Some of the weaknesses of the SCE and SPA identified by the study include the following:

• During source selection, only existing, in-place processes were evaluated by the SCE team. Software engineering process improvements proposed by the contractor for the software development project had little bearing on the SCE results. There may be important differences between the software engineering processes being proposed for a specific acquisition and the processes being evaluated by the SCE team. [Ref. 16:p.20]

• Several areas important to software development are not covered in the CMM, and therefore, are not evaluated in the SCE. These areas include human resources, software engineering methods and tools, and product specific technologies. [Ref. 16:p. 20]

• Results between SCEs and SPAs can differ as much as one or two maturity levels. The difference can be attributed to differences in interpreting SEI rating criteria, differences in type of projects, and differences in SPA and SCE team experience. [Ref. 16:p. 36]

The IDA study team reviewed the SDCCR and compared it with the SCE. The IDA reported the following as strengths of the SDCCR:

• The SDCCR method focused on the software development processes and capabilities proposed for a specific acquisition program.

• The SDCCR used a set of essay questions as one means of obtaining information from the contractor. This allows an evaluation to occur when no "discussions" are allowed; that is, the winner of the contract will be based solely on the evaluation of the proposals submitted by the offerors. Contractors also report that although answering the questions is costly, the exercise provides valuable insight to the issues important to the Government and into their own development practices.

• The SDCCR site visits allowed contractors to explain and rationalize their proposed software development approach.

• The SDCCR evaluated areas not addressed by the SCE such as systems engineering. [Ref. 16:p. 95]

The report identified the following weaknesses of the SDCCR:

• The SDCCR evaluation team relied on an offeror's documentation to verify that the proposed practices have been used on previous projects. No individual interviews were conducted to verify that the documents accurately reflect the processes being used.

• The SDCCR contained 450 essay questions. It could cost a contractor as much as $500,000 to answer the SDCCR questions.

• The criteria for scoring or rating answers to the SDCCR questions were not completely defined.

• The SDCCR results provided limited input towards process improvement.

• There were no guidelines to tailor the SDCCR to a specific acquisition. [Ref. 16:p.95]

It is important to note that these studies were based on older versions of the SCE and CMM [Ref. 16:p. 5]. They also reflect the strengths and weaknesses of the SCE and SDCCR, both of which formed the baseline for developing the SDCE.

In the interviews conducted by the researcher, the discussions on the strengths and weaknesses of the SCE and SDCE centered around these same issues mentioned by the studies. In an interview with the researcher, an SEI representative raised the additional issue concerning the uncertainty surrounding the newly developed SDCE method versus the proven benefits of the SCE.

C. SUMMARY

This chapter presented the findings of the Aerospace Corporation and IDA studies on software capability evaluations. Strengths and weaknesses can be categorized in the following areas: scope of the evaluations, credit for new processes, questionnaires, tailoring to a specific acquisition, organization versus project focus, conducting site visits, process improvements, possible sampling errors, possible differences between SPA and SCE, and history of success. These areas are explained in the following chapter. The next chapter also compares the SCE and SDCE in these areas.

VII. COMPARISON AND ANALYSIS

In this chapter a comparison of the SCE and SDCE in the areas identified in the previous chapter is made. The issues involving each area, data pertinent to the issues, and conclusions will be discussed.

A. COST OF THE EVALUATIONS

1. Issue

PMs have finite resources. An issue in deciding which evaluation to use is the cost to conduct an SCE or SDCE. SEI and ASC/AFMC measure the cost of their respective evaluations in terms of human resources, or person days.

2. Research Findings

Figure 14 depicts SEI's estimate to conduct an SCE for a single source selection with three offerors within the competitive range. The estimated cost is approximately 115 person days.

ASC/AFMC's estimate to conduct an SDCE for single source selection with three offerors within the competitive range is also shown in Figure 14. The estimated cost ranges from 200 to 314 person days.

These figures show that the cost of conducting an SDCE is almost double or triple the cost of conducting an SCE. This disparity in costs can be attributed to the following:

SCE Phase	Effort (Person Days)
SCE Information Gathering	5.25
RFP Preparation	3.0
SCE Training	25.0
SCE Findings Preparation	5.0
Site Visits	75.0
Contractor Debriefs	1.5
SCE Total for Three Site Visits	114.75
SDCE Activity	**Effort (Person Days)**
SDCE team preparation through final RFP release	20 - 80
SDCE team proposal analysis prior to site visit	54 - 72
SDCE team site visit	72 - 90
SDCE team evaluation after site visit	54 - 72
SDCE Total for Three Site Visits	200 - 314

SCE Source: CMU/SEI-93-TR-18
SDCE Source: AFMC Pamphlet 800-61

Figure 14 SCE Versus SDCE Manpower Costs

- The SCE team is composed of four to six personnel [Ref. 5:p. 22]. An SDCE team is composed of a core and support team. The core team has four personnel; the support team could have up to eight. [Ref. 3:pp. 50-51] An SDCE team could therefore be double or triple the size of an SCE team.

• Figure 14 shows that SDCE teams expend a lot of effort on proposal analysis. The SDCE uses essay type questions, which provide more contractor information to evaluate than the SCE "yes/no" questions.

• Figure 14 also shows that an SDCE team applies a considerable amount of labor for evaluation activities after the site visits have been conducted. SCE activities end once the SCE team delivers its findings to the SSEB. For the SDCE, core team members participate in the evaluation of all offerors as members of the SSEB.

The figures provided above are estimates. Many additional factors can affect the actual costs. These include the complexity of the system being developed, the complexity of the source selection, the team's prior experience in conducting software capability evaluations, the number of people on the team, and the personnel used on the team (contractor, Government civilian, or military) [Ref. 3:p. 62] [Ref. 10:pp. 3-54 - 3-56].

3. Conclusions

The findings show that due to the greater volume of information to analyze prior to the site visits, the potential for an SDCE team to be double or triple the size of an SCE team, and participation of SDCE team members in the SSEB, the cost of conducting an SDCE is significantly greater than that of an SCE.

B. SCOPE OF THE SDCE AND SCE

1. Issue

Mission Critical Computer Resources (MCCR) refers to the totality of computer hardware and software that is integral to a weapon system, along with the associated

personnel, documentation, supplies, and services [Ref. 1:p. 1]. As the definition of MCCR implies, there are many critical areas that affect software development. These include such issues as systems engineering, human resources, tools, and technology. An issue a PM must consider is whether the scope of each evaluation method is broad enough to cover these areas.

2. Findings of Previous Studies

According to the IDA study, the SDCCR covered areas not addressed by the SCE to include systems engineering and tools [Ref. 16:p. 95]. In the interview mentioned earlier, an author of the Aerospace Corporation study came to the same conclusion.

3. Research Findings

The SDCE inherits the broad scope of the SDCCR. In fact, the SDCE model is composed of critical capabilities that have historically been high-risk areas during software development [Ref. 3:p. 12]. For example, it addresses the systems engineering activities that impact the software development process.

Systems Engineering is important in the development of a weapon system. The Defense Systems Management College defines systems engineering as follows:

> Systems Engineering is a management function which controls the total system development effort for the purpose of achieving an optimum balance of all system elements. It is a process which transforms an operational need into a description of system parameters and integrates those parameters to optimize overall system effectiveness. [Ref. 17]

It is through systems engineering that software requirements are developed.

Problems occur when software requirements do not match the system's requirements. The case of the Cheyenne Mountain Upgrade (CMU) development can serve as an example of this point. The CMU program is the modernization of command and control subsystems at Cheyenne Mountain, which serves as the command center for the North American Aerospace Command (NORAD). These subsystems provide critical strategic surveillance and attack warning to US and Canadian leaders. [Ref. 18]

There were two systems engineering related software problems with CMU. First, because the USAF did not adequately define the system's requirements, software was developed for a specific hardware platform. This presents a significant software portability problem should the USAF decide to upgrade the computer hardware. Also, because of the lack of adequate system requirements, the subsystems are not expected to properly interface with each other. [Ref. 18]

Unlike the SCE, the SDCE places heavy importance on systems engineering. Those areas of systems engineering that affect software development are included in the SDCE model in the Systems Engineering FA. A senior systems engineer is part of the core evaluation team. Finally, emphasis is placed on comparing software development processes with key systems engineering documents such as the SEMP, SEDS, and SEMS.

Human resources is another area addressed by the SDCE and not by the SCE. Human resources are important because software development is largely labor intensive. The outcome of a project depends on the abilities of the development team's personnel. The Human Resources CCA addresses an offeror's capability to recruit, retain, and efficiently allocate skilled labor to support the software development. [Ref. 3:p. 35]

Another area of concern in the development of embedded software is specific technologies required by the project. This is illustrated by the development of the BSY-2 software. The BSY-2 is a combat weapon system designed to detect, track, and launch weapons for the US Navy's new attack submarine, the Seawolf. One of the software problems identified by GAO on the BSY-2 was the incomplete data base management, design which may degrade system performance [Ref. 19]. The CCA Data Base Management under the Program Specific Technologies FA of the SDCE model evaluates an offeror's ability to develop a complex data base system. [Ref. 3:p. 39]

A project's unique technology may not be in the Program Specific Technologies FA of the SDCE model. The SDCE method, however, allows the PMO to develop questions and evaluation criteria for a specific technology and add it to the evaluation.

The SCE evaluates a contractor's process capability against the CMM. The CMM was designed to focus on a limited set of key process areas that have been proven to enhance software development and maintenance capability. As a result, the CMM, and therefore the SCE, does not address important areas such as systems engineering, human resources, and specific technologies.

The SEI acknowledges that there are important issues outside software development that are not addressed by the SCE. The SEI states that these areas should be evaluated, but done outside the context of the SCE [Ref. 10:p. 1-19].

In order to evaluate these important areas, many SCE users develop pseudo KPAs. For example, in the course of conducting an SCE to evaluate offerors for a $95 million

software avionics contract, the SCE team noted that the SCE did not address systems engineering issues [Ref. 20].

> We must investigate key issues in hardware and systems engineering in our contractor selection because few of our projects involve only software development. Most of our application software integrates with hardware and is a modification or enhancement of existing software.

The team created a hardware and systems engineering questionnaire to add these areas to the evaluation.

Another example was provided in an interview between the researcher and an SCE user who had conducted four SCEs for the SDI program. She stated that the SDI program has special security requirements that an offeror must meet. These security requirements are not covered in the CMM, and therefore not evaluated by the SCE. She developed "pseudo" KPAs using the National Security Agency's (NSA) Trusted Methodology method.

The limited scope of the SCE is tied to the CMM. As stated in Chapter II, the SEI may expand the CMM in future versions. The researcher also discovered in an interview with an SEI representative that the SEI is working with industry and professional organizations to develop a Systems Engineering CMM and a People CMM. Ownership and maintenance responsibilities for these CMMs have yet to be established.

4. Conclusion

The SDCE team evaluates an offeror's software development capability against the SDCE model. The SDCE model is composed of critical capabilities that have historically been high-risk areas in past software developments. The SDCE model address areas not

found in the CMM such as systems engineering, human resources, and specific technologies. As a result, the SDCE has a greater scope of evaluation than the SCE.

C. CREDIT FOR NEW PROCESSES

1. Issue

Both methods review software development documents to verify that processes and procedures are actually implemented in accordance with an organization's documentation. An issue that a PM must consider is how does the SDCE and SCE deal with new processes proposed by an offeror that has little or no documented history.

2. Findings of Previous Studies

The IDA study criticized the SCE for only evaluating existing processes. Process improvements newly initiated by a contractor had little or no impact on the SCE results because proof of implementation did not exist. This criticism was echoed by an author of the Aerospace study.

3. Research Findings

As stated in Chapter III, literature research indicates that this weakness has been corrected in the SCE version 1.5. SCE results have been changed to reflect the strengths, weaknesses, and process improvement activities at the KPA level. Ongoing process improvements related to the KPAs of the target process capability are evaluated during the site visit. The SCE team uses its knowledge and experience to determine and report the adequacy of ongoing process improvements to the SSEB.

The SDCE method also evaluates and reports on new processes. When a new process is proposed by the contractor for use on the project, because there is no proof of past implementation, a CR is generated. This new process then becomes a topic of discussion during the site visit. The SDCE team evaluates whether the new process is appropriate to the project and whether the contractor can sufficiently execute it as planned.

4. Conclusion

The importance of evaluating and reporting new processes is best stated by the ASC/AFMC representative who said, "If we did not give credit to new processes, we would still be coding software using ones and zeroes." Both methods evaluate and report new processes. It is important to note an important difference between how the SDCE and SCE address new processes. As mentioned previously, each method only evaluates new processes that are covered in their respective models. The SDCE model addresses areas not covered by the CMM such as systems engineering, human resources, and specific technologies. In a SDCE evaluation, a contractor can receive credit for new processes in these areas, where it could not in an SCE.

D. QUESTIONNAIRES

1. Issue

Both methods use a questionnaire to provide information concerning an offeror's software development capability. An issue a PM should consider is what are the strengths and weaknesses of the SDCE and SCE questionnaire.

2. Findings of Previous Studies

According to the IDA study, a strength of the SDCCR was its essay type questions. The SDCCR questions solicited more information on an offeror's software development capability than the SCE questionnaire. This allowed the SDCCR to be used in source selection evaluations under the "without discussions" condition. In "without discussions" source selections, site visits may not be conducted. However, a limited evaluation of an offeror's software development capability could be conducted based on an offeror's responses to the SDCCR questions.

The IDA study identified additional benefits of the SDCCR questionnaire to contractors as well. The SDCCR questions used in the source selection help key contractors on the Government's concerns regarding the acquisition. Contractors also report that answering the questions provides valuable insight into their own processes and capabilities that will be used on the project.

According to the IDA study, there was a problem with the SDCCR questions. The criteria for scoring or rating the answers to the SDCCR questions were not completely defined.

The author of the Aerospace study found the questionnaire used by the SCE to be a weakness. The "yes/no" response to the SCE questionnaire did not provide the detailed information required to determine the adequacy of the process.

3. Research Findings

The researcher questioned the ASC/AFMC representative about the SDCCR's role in a "without discussions" source selection. He stated that many source selection plans

for weapon system acquisitions are written to award the contract "without discussions." Although these acquisitions begin with the "without discussions" condition, because weapon system acquisitions are complex, the SSA usually opens the source selection to allow "discussions" between the Government and the offerors.

This is not always the case, however. A contract for an F-16 component was awarded "without discussions." The SDCCR was used during this acquisition's source selection activities to evaluate an offeror's software development capability. Since "discussions" were not allowed, no site visits were conducted. The offerors' software development capabilities were evaluated by their answers to the SDCCR questions and the content of their software development plans.

In an interview with the researcher, a contractor representative stated that his company has been subjected to both the SDCCR and SCE. Many in his company found it easier to answer the SDCCR questions than those of the SCE. The company could describe how a process is performed within the organization in response to a SDCCR question. He explained that the problem with answering the SEI maturity questionnaire stems from an unclear definition of a "yes" or "no" answer. For example, a question from the SEI maturity questionnaire asks "Does each software developer have a private computer-supported work station/terminal? If 99% of the workers have a terminal, does this call for a 'yes' or 'no'?" The contractor representative also stated that the exercise of answering the SDCCR questions provided insight into the company's software development capability.

As stated in Chapter IV, the SDCE questions are in the same essay format as the SDCCR. The essay type questions makes the SDCE method useful during "without discussions" source selections. The questions are part of the SDCE model.

As mentioned in Chapter IV, and illustrated in Figure 5 of the same chapter, model criteria are defined for each SDCE question. This solves the problem identified by the IDA study concerning the absence of defined criteria to score the SDCCR questions.

As mentioned in Chapter III, the SEI acknowledges that there were problems with its maturity questionnaire. The questionnaire does not address some KPAs. Some questions are not linked to any KPA at all. The SEI has mitigated the questionnaire problem by switching the SCE focus from being "question-based" to "model-based." According to the SEI, the maturity questionnaire is used only as an input source during the specific preparation phase and does not have a direct impact on the SCE results [Ref. 10:p. 2-39]. This was verified by the SCE user supporting the SDI program in an interview. She said that the questionnaire is hardly used. When conducting site visits, emphasis is placed on verifying KPAs, not the offeror's response to the questionnaire.

As mentioned in the previous paragraph, the purpose of the SEI's maturity questionnaire during an SCE is to provide input to the specific preparation phase and has little impact on the SCE results. As stated in Chapter III, the SCE uses two primary means of soliciting information to evaluate an offeror's process capability. They are interviews and documentation reviews, not the maturity questionnaire. In addition, the offerors record their answers to the maturity questionnaire on an SEI assessment-recording form. In this form, each question has a section that solicits comments from offerors to

amplify their "yes/no" responses. [Ref. 22:pp. 39, 44] Based on this information, SCE version 1.5 eliminates the criticism that the binary format of the SEI maturity questionnaire does not provide enough information to determine the adequacy of the process.

The SEI released a new questionnaire based on the CMM version 1.1. this year. In an interview, an SEI representative stated that the new questionnaire has corrected the problem of relating KPAs to questions experienced by the previous version of the maturity questionnaire.

4. Conclusion

The SDCE essay type questionnaire solicits more information than the SEI's maturity questionnaire. Of the two methods, only the SDCE is useful in "without discussions" source selections. The process of answering the SDCE questions provides an offeror with insight into its own software development capability. The criteria for scoring the SDCE questions are also defined.

There were problems of relating questions to KPAs in the previous version of the maturity questionnaire. The SEI has released a new maturity questionnaire that has corrected these problems.

E. TAILORING

1. Issue

As mentioned in the previous chapters, each method uses a tailored subset of their respective model to evaluate an offeror's software development capability for a specific

acquisition. An issue PMs must address is the ease with which the SCE or SDCE may be tailored to address the unique requirements of their programs.

2. Findings of Previous Studies

A weakness of the SDCE's predecessor, the SDCCR, was that no guidance was provided for tailoring the SDCCR for a particular acquisition [Ref. 16:p. 95]. In an interview, a member of the IDA team that conducted the study stated that some SDCCR teams found it too difficult to tailor the SDCCR. As a result, the SDCCR teams sent all 450 questions for the offerors to answer. It was estimated that it could cost a contractor up to $500,000 to answer all the SDCCR questions [Ref. 16:p. 46].

The author of the Aerospace Corporation echoed the same criticism in her interview with the researcher. She stated that the structure of the SDCCR model made it difficult to tailor the method to a particular acquisition.

3. Research Findings

In the same interview with the author of the Aerospace Corporation study, she stated that she was a member of the SDCE model development team that was part of the SDCE PAT. She has also used the SDCE model and questionnaire four times in source selections for aircraft radar and satellite systems. She found that it was easy to tailor the SDCE model for each acquisition. The manner in which the SDCE model is organized fixed the problem the SDCCR had of tailoring.

The researcher questioned the ASC/AFMC representative on the use of the SDCE model in these acquisitions. He stated that he knew of these acquisitions. In them, the

Aerospace Corporation evaluation team did not complete all the activities of the SDCE method. Therefore, his office does not recognize these acquisitions as pilot uses of the SDCE method.

In the same interview, the ASC/AFMC representative stated the SDCE model structure facilitates the tailoring of the SDCE method to a specific acquisition. For example, eliminating a FA automatically eliminates the model questions of the CCs associated with the FA.

The ASC/AFMC representative also provided another explanation as to why the entire SDCCR questionnaire was sent to the offerors. He stated that the questions cover so many important risk areas that SDCCR teams wanted answers to all the questions. He is encountering this problem in the pilot use of the SDCE method on the TSSM program. According to the ASC/AFMC representative, the TSSM PMO originally identified a large number of CCs for use in the source selection evaluation of an offeror's software development capability. These CCs encompassed approximately 800 questions from the SDCE model. The ASC/AFMC office is working with TSSM PMO to reduce the number of SDCE questions for this acquisition.

As described in Chapter III, the SCE has an effective method of tailoring. It uses the maturity levels. Once the SCE team has determined the target process capability, the KPAs of the corresponding maturity level and the maturity levels below it are selected for evaluation.

4. Conclusions

The SCE is the easiest of the two methods to tailor to a particular acquisition. Tailoring is based on the target process capability and maturity levels. Although the Aerospace Corporation's use of the SDCE model is not recognized by ASC/AFMC as a pilot use of the SDCE, it shows that the problem of tailoring associated with the SDCCR model structure has been corrected in the SDCE model. A problem facing SDCE users now is limiting the CCs and associated questions for use in a SDCE evaluation to those high value discriminators that will provide the greatest insight into an offeror's software development capability.

F. PROJECT FOCUS

1. Issue

The PM is responsible for the success of his program. His primary focus is on those factors affecting his program. An issue a PM should consider when choosing between the two methods is which one has a greater focus on the software development risks of his program.

2. Findings of Previous Studies

According to the IDA study, a strength of the SDCCR method was that it focused on the proposed processes and capabilities that were proposed by an offeror to develop the software project. It did this by scrutinizing the software development plan submitted by an offeror with its proposal. [Ref. 16:p. 46]

3. Research Findings

As mentioned in Chapter V, the SDCE, like the SDCCR, focuses on the processes and capabilities that are required to successfully develop the software for the specific project. Based on a project's unique characteristics, the SDCE team identifies those CCs that are crucial to the success of the software development project. The vendor is evaluated on its ability to perform these CCs.

Members of the software industry and Government have criticized the SCE for focusing more on organizational processes and capabilities rather than those required at the project development team level [Ref. 3:p. 12] [Ref. 13:p. 1]. For example a concern of the Aerospace Industries Association (AIA) is:

> While the Government must validate that a company uses the processes it professes, it must also focus the examination on how policies, practices and procedures would be implemented in the project under evaluation, with attention to what is unique on a program and to a selected contractor's entire development team. [Ref. 12:p. 2]

As illustrated in Chapter III, the SCE version 1.5 does focus on the processes required for the proposed software development, but within the context of the CMM and the target process capability. Based on the project's requirements, the SCE team determines a target process capability an organization must possess to successfully develop the project's software. An offeror is only evaluated on the KPAs of target process capability and the maturity levels below it.

4. Conclusions

Both methods focus on the processes and capabilities required to develop a particular project's software. As discussed earlier in this chapter however, the SDCE has

a greater scope than the SCE. The SDCE therefore focuses more on the processes and capabilities required to develop a project's software.

G. CONDUCTING SITE VISITS

1. Issue

Each method uses a different approach in conducting site visits. An issue a PM should consider is which of the two approaches is better.

2. Findings of Previous Studies

The IDA study identified that the lack of individual interviews was a weakness of the SDCCR. Individual interviews are required to verify that the documentation accurately reflects the processes being used.

The results of the survey used in the study indicated that 68 percent of the contractors who have been subjected to more than one SCE believed that SCEs are a useful method for the Government to select and monitor contractors.

The IDA study also identifies the SDCCR site visit as a strength of the method. "The SDCCR includes contractor site visits, which allow contractors to explain and rationalize the software development approach they have proposed for the acquisition program." [Ref. 16:p. 46]

3. Research Findings

There are differences in the way site visits are conducted between the two methods. This stems from the different views taken by the SDCE and SCE on the purpose of the site visit.

As discussed in Chapter III, the site visit allows the SCE evaluation team to collect information on the offeror's process capability through interviews and documentation review. Personnel at the organizational level down to the development teams of those projects selected for review are interviewed individually by the entire SCE team. At first glance, this setting may resemble an interrogation, but SCE teams are trained to place the interviewee at ease.

As one SCE user described in an interview, the SCE team interviews an individual on a topic or several topics. The interviewee's responses are then compared with documentation and information collected from other interviews. Discrepancies may show that processes do not exist, or may be established but are not followed or understood.

For the SDCE, one of the goals of the site visit is not only to address software development issues, but to explore these issues in a "positive, team building atmosphere" or non-adversarial manner [Ref. 3:p. 89]. Specifically, a contractor representative, who was a member of the SDCE PAT, said that the SDCE was designed to avoid the interrogation-like atmosphere of the SCE.

One of the purposes of a SDCE site visit is to provide "a forum for the SDCE team and an offeror to discuss the proposed capability and processes, in an open dialogue, with the objective of reaching a mutual understanding of the offeror's capability in terms of processes, resources, experience, skills, tools, and technology" [Ref. 3:p. 88]. Because of this, the offeror selects the team members who will represent the company during the site visit, not the SDCE team. Also, before leaving the offeror's site, the SDCE team

presents its understanding of the proposed development processes and capabilities, then gives the offerer a chance to respond.

There are no data to suggest that one site visit approach is better than the other. As mentioned previously, the SEI claims that the SCE site visit is a proven and effective method of investigation in its publication. In an interview with the researcher, the ASC/AFMC representative claimed the SDCCR/SDCE site visit method is also effective. He stated that an experienced SDCE team can quickly discern whether a process is adequate during the site visit. He pointed to the successful use of the SDCCR on over 30 weapon system source selections. For example, the SDCCR was successfully used on several source selection evaluations for contracts dealing with the USAF's next generation fighter, the F-22.

As discussed in previous chapters, two common factors to both site visit methods are the site visit planning and the selection of the evaluation team. Effective site planning identifies the areas to be evaluated, the information required, the documentation to request, and the questions to ask. This maximizes the amount of useful information received during the short duration of the site visit. Staffing the evaluation team with knowledgeable and experienced personnel allows the team to conduct an effective evaluation whether using individual interviews or a discussion format.

4. Conclusion

When properly staffed with knowledgeable personnel, the SCE or SDCE team can obtain information on an offeror's software development capability. The strength of the SDCE site visit is that it allows the offeror a chance to respond to the SDCE team's

findings to ensure information provided by the offeror was not misinterpreted or missing. The strength of the SCE site visit is that it conducts individual interviews to verify that the documentation accurately reflects the processes being used, and to evaluate the level of understanding key personnel have of the software development process.

H. PROCESS IMPROVEMENT

1. Issue

The effects of the SDCE and SCE on the software industry's process improvement activities is also an issue.

2. Findings of Previous Studies

The IDA study found the SCEs were useful in promoting desired process improvement. IDA supports this finding by the survey results. As mentioned previously, 68 percent of the organization who were subjected to more than one SCE believed that SCE are useful in identifying the strengths and weaknesses of a contractor software development process. The SCE findings could be used as input to an organization's process improvement plan.

3. Research Findings

As mentioned in Chapter III, one of the benefits of the SCE is that it promotes software process improvement. As Government agencies use the SCE during source selection, contractors must adopt the CMM based process improvement plans in order to stay competitive for Government contracts. Because the SCE is based on the CMM,

offerors have a defined method of allocating scarce resources to process improvement activities, in order to do better in future evaluations.

An example of this is the Equipment Division at Raytheon. It used the CMM and SPA to implement process improvement to increase its maturity level from two to three. Raytheon credits its high maturity level as the deciding factor in winning two Government contracts. With the increased use of SCEs by the Government, Raytheon has placed great emphasis on process improvement. [Ref. 21]

A concern of the software industry is that as more evaluations are conducted on an organization, process improvement plans might become less stable as the company constantly reprioritizes resources in an attempt to fix the deficiencies found from the latest evaluation [Ref. 12:p. 2]. This is not a problem for those offerors using CMM-based process improvement activities when being evaluated by the SCE. Since the SCE evaluates how well an offeror conforms to the CMM, the results could be viewed as an outside assessment of the company's process improvement activities. The SCE results can easily be incorporated into the organization's process improvement plan.

Promoting process improvement is not a stated benefit of the SDCE [Ref. 3:pp. 4-5]. The SDCE is not tied to any process improvement methodology. Although the SDCE incorporated aspects of the CMM, the maturity levels were eliminated. With the elimination of the maturity levels, an offeror has no way of prioritizing allocation of resources to prepare for the SDCE. With the SDCE model containing 117 CCs incorporated into 37 CCAs, which in turn are grouped into six FAs, it is difficult for a contractor to prioritize process improvements to do better during an SDCE.

An SEI representative told the researcher in an interview that he was concerned that the SDCE model might reverse the process improvement gains realized in the software industry. The CMM is a proven model in assisting an organization in obtaining higher levels of process maturity. Contractors may be tempted to abandon process improvement activities in favor of satisfying the requirements of future SDCE evaluations.

During an interview with a contractor representative who served as a member of the SDCE PAT, it was stated that the SDCE should have adopted the CMM maturity levels. His company does business with the other Services, not just the USAF (the only Service to use the SDCE). In order to stay competitive, his company must pursue process improvements using the CMM. In his opinion, the SDCE should have taken the SCE as a base and added other areas such as systems engineering. It would have been easier for his company to contend with just an expanded SCE than two different evaluations.

Although the offeror is encouraged to incorporate SDCE results into its process improvement plans, this task may not be easily done. As previously discussed, the SDCE model contains areas that are not addressed in the CMM, and deficiencies in these areas may not be readily incorporated into an organization's process improvement plans. With the elimination of the maturity levels, an area identified as weak by the SDCE may be a KPA that is above the organization's maturity level as denoted by a CMM/SPA.

In an interview with the researcher, the ASC/AFMC representative said that the SDCE was designed strictly as a capability evaluation, not as a method of promoting process improvement. As an analogy, he indicated that the SEI's SPA was not an evaluation but a process improvement tool. While the SDCE method does examine an

organization's process improvement activities, it only evaluates those being used on the project.

4. Conclusion

A strength of the SCE is that it promotes process improvement within the software industry. The SDCE is strictly an evaluation tool and was not designed to promote process improvement.

I. SAMPLING ERROR

1. Issue

Another AIA concern regarding the SCE is the application of a single maturity rating to the various divisions or groups within a corporation when multiple projects are evaluated. The issue being raised is the accuracy of the SCE results based on an evaluation of a sample size of three or four software projects. [Ref. 12:p. 2]

2. Research Findings

The SCE team draws conclusions on an organization's software development process by examining three or four of the organization's completed or ongoing projects. According to the SEI, "Using its collective professional judgment and a consensus decision making process, the SCE team puts together its findings from individual projects to create a set of overall, organization-level findings" [Ref. 10:p. 2-40].

One interviewee who had conducted four SCEs stated to the researcher that a disadvantage to the SCE is that only a small sample of an offeror's projects are examined.

KPAs may be evaluated as weak for these projects, but may indeed be well-implemented throughout the organization overall.

The SEI is aware of this possible shortcoming and provides the following guidance to SCE teams:

> Findings should be specific to the point where they identify the cause for a strength or weakness, but not so specific that the finding places the team in a corner by failing to consider exceptions that may exist within an organization. SCE teams must remember they are evaluating a subset of the total projects ongoing at a site as a proxy for predicting the organization's capability to do a specific project, and exceptions may exist because of this process. [Ref. 10:p. 2-40]

The SCE method heavily depends on the SCE team members' experience and knowledge to accurately estimate the process capability of an organization based on the small sample of projects being evaluated. However, there exists a possibility that the software development capabilities demonstrated by the projects that were evaluated do not represent the organization as a whole.

3. Conclusions

A potential weakness of the SCE method is that the team draws a conclusion on the software development capability of an organization by evaluating three to four of the organization's completed or ongoing projects. While these conclusions may reflect the process capability of the projects that were evaluated, it may not necessarily represent the entire organization. The organizational software development capability may actually be stronger or weaker than those of the projects evaluated by the SCE team.

J. DIFFERENCES IN SPA AND SCE RESULTS

1. Issue

Another issue concerning the validity of SCE results stems from possible differences between the SPA and SCE results.

2. Findings of Previous Studies

The IDA study found that there were differences between SCE and SPA results. For example, the USAF reviewed two source selections encompassing 14 contractor site visits. The SCE findings indicate that approximately 20 percent of the contractors were a Maturity Level 3 as compared to the contractors' SPA based claims that approximately 60 percent were Maturity Level 3 organizations [Ref. 16:p. 36]. The IDA team attributed these differences "to many reasons: different interpretations of the SEI rating criteria, different projects being evaluated, different levels of experience between teams, and so forth." [Ref. 16:p. 36]

3. Research Findings

The SEI recognizes that differences between the SCE and SPA results can occur and emphasizes the causes of this problem in its publications. For instance, the SEI assumes that contractor personnel have no motive to mislead SPA team members who, with the full support of management, are dedicated to process improvement to raise the competitiveness of the company. In a SCE, the SEI assumes that the evaluated organization will make every attempt to put its process capability in the best possible

light. SCE results could therefore be higher. The SCE team must make every attempt to verify the facts concerning an offeror's process capability. [Ref. 10:p. 1-21]

Much of the evidence to support the CMM and process maturity is based on the positive results experienced by organizations as they improve to the next maturity level. In these cases, improvements in maturity level were verified using SPA not SCE results. The Software Engineering Division of Hughes Aircraft serves as an example [Ref. 22]. Hughes Aircraft's Software Engineering Division reported a cost savings of two million dollars a year by improving from Maturity Level 2 to Maturity Level 3.

A problem arises when SCE results differ from SPA results. Suppose an SCE was performed on the Software Engineering Division of Hughes Aircraft and the results indicate that the organization was at Maturity Level 2. One could conclude that the SPA results were wrong, and therefore an organization need not improve to a higher level of maturity to realize significant cost savings. This could cast doubt on the CMM and process maturity concept. One could also conclude that the SCE results were wrong, thereby presenting an inaccurate picture of Hughes SED's software development capability.

To address the problem of dissimilar results for the SPA and SCE, the SEI has combined the SCE and SPA projects into one appraisal project based on the CMM. The CMM Based Assessment (CBA) project will develop a common rating framework to describe an approach to determine and report process maturity for all CMM based appraisals. It will also provide guidance and standards for performing appraisals to ensure

the accuracy and consistency of results. The framework will have two algorithms: one for determining the extent to which an organization or project satisfies each subprocess area and another to determine an organization's maturity level. SEI has not published a schedule of when these products will be available. [Ref. 23]

In an interview with the researcher, a SEI representative stated that the SEI no longer conducts training on the SCE method. All SCE training is being conducted by a contractor. The SEI maintains ownership of the SCE method and training to ensure SEI standards are maintained.

4. Conclusion

The SEI acknowledges that SCE results may differ from those of an SPA even though they are both based on the CMM. When this occurs, the validity of the SCE results may be questionable. The SEI has created the CBA project to correct this problem.

K. RECORD OF SUCCESS

1. Issue

As stated in Chapter I, modern weapon systems rely heavily on their computer hardware and software to operate in a manner in which they are designed. An issue important to the PM is which software capability evaluation has a proven history of success.

2. Findings of Previous Studies

The IDA study cited anecdotal evidence of organizations experiencing significant positive ROI by increasing their process maturity levels. These cases were used to support the pursuit of process improvement based on the CMM. They were the following:

- Hughes Aircraft's Software Engineering Division reported a decrease in development risks as it moved from Maturity Level 2 to Maturity Level 3. Hughes used the Cost Performance Index (CPI) as an indicator of risk costs. CPI is calculated by dividing the budgeted cost of work performed (BCWP) by the actual cost of work performed. In July 1987, CPI was .94, which means actual costs exceeded budgeted costs by 6 percent. After implementing process improvements, CPI had increased to .97. This indicates that actual costs exceeded budgeted costs by only three percent, a drop of 50 percent. The estimated cost savings to Hughes was two million dollars annually. [Ref. 22]

Ratheon's Equipment Division also used the CMM and SPA to increase its maturity level from two to three. It reports the elimination of rework costs at a saving of $15.8 million. Many software projects are being delivered below budget and ahead of schedule. In one case, this earned Raytheon a $9.6 million schedule-incentive payment. Raytheon also reports a return on investment (ROI) of 7.7 to 1 (a $4.48 million return on a $0.58 million investment) for implementing process improvement. [Ref. 21]

3. Research Findings

As stated in Chapters II and III, the SCE and CMM were developed in the late 1980s. The current versions of the SCE and CMM represent approximately eight years of continuous improvements from their original versions. These improvements are based on lessons learned and feedback from the Government and software industry. In an interview, the SEI representative said that the CMM is one of the most widely used models in the software industry worldwide.

As stated in Chapter IV, the SDCE was developed by combining the strengths and correcting the weaknesses of two proven evaluation methods, the SDCCR and SCE. The model used by the SDCE is a collection of critical capabilities that have historically been areas of high risk in software development. While these two factors may indicate that the SDCE is an effective means of evaluating an offeror's software development capability, the SDCE method is still unproven. Pilot tests of the SDCE are scheduled for next fiscal year.

4. Conclusions

The current versions of the SCE and CMM have a proven history of success. Although the SDCE has the potential for being an effective method for evaluating an offeror's software development capability, it is still unproven. It will continue to remain so, until the pilot tests of the SDCE are completed next fiscal year

L. Summary

This chapter addressed the issues concerning the use of the SDCE and SCE in the source selection of a software intensive weapon system. Based on the previous studies and the findings of this research, conclusions were developed for each area. The following chapter will summarize these conclusions.

VIII. SUMMARY AND RECOMMENDATIONS

In the previous chapter, conclusions were developed from the research findings in various areas. This chapter uses these conclusions to answer the research questions. The secondary research questions are addressed first, followed by the primary question. Recommendations will be made. And finally, areas of further research are identified.

A. ANSWERS TO THE SECONDARY RESEARCH QUESTIONS

All the secondary research questions have been answered in previous discussions. They are summarized in the following paragraphs.

1. What is the Capability Maturity Model?

This question was answered in Chapter II. The CMM represents a limited collection of KPAs that have been shown to enhance software development and maintenance capability. By focusing on this limited set of KPAs and working aggressively to achieve them, an organization can steadily improve its process capability. The CMM is a tool to assist developers in gaining control of their software development process and move towards continuous process improvement.

2. What is a Software Capability Evaluation (SCE) and how is it used during source selection?

This question was answered in Chapter III. The SCE is a method developed by the SEI for evaluating the software development process of an organization. The SCE's role

in the source selection is to support the acquisition of software by assessing an offeror's software process capability. The SCE identifies the strengths, weaknesses, and process improvement activities of an offeror. The SCE results are incorporated into the source selection evaluation.

3. What is the Software Development Capability Evaluation Model?

This question was answered in Chapter IV. The SDCE model is a collection of critical capabilities that have historically been high-risk areas during software development. The model contains the evaluation questions and criteria for each critical capability.

4. What is the Software Development Capability Evaluation and how is it used during source selection?

This question was answered in Chapters IV and V. The SDCE is a method used during source selection to evaluate an offeror's software engineering and management capabilities. It also evaluates an offeror's systems engineering capabilities which directly impact software development. The primary purpose of the SDCE is to increase the probability of selecting an offeror capable of successfully developing software that meets all requirements within cost and schedule constraints.

B. THE PRIMARY RESEARCH QUESTION

The primary research question is "What are the strengths and weaknesses of the SEI's Software Capability Evaluation and ASC/AFMC's Software Development Capability Evaluation, that PMs should consider when deciding which method to use on

their programs to evaluate a contractor's software development capability during source selection?". To answer this question, the conclusions that were analyzed and discussed in depth in the previous chapter will be categorized as strengths or weaknesses for each method. A summary of the strengths and weaknesses of each method is depicted in Figure 15.

SDCE	SCE
STRENGTHS * LARGER SCOPE * ESSAY-TYPE QUESTIONS * PROJECT FOCUS * SITE VISIT	**STRENGTHS** * LOWER COSTS * INDIVIDUAL INTERVIEWS * PROCESS IMPROVEMENT * EASIER TO TAILOR * PROVEN METHOD
WEAKNESSES * MORE EXPENSIVE * DIFFICULT TO TAILOR * PROCESS IMPROVEMENT * UNPROVEN METHOD	**WEAKNESSES** * SMALLER SCOPE * RESULTS BASED ON SMALL SAMPLES * DOES NOT MATCH SPA

Figure 15 A Summary of Strengths and Weaknesses

1. SDCE Strengths

This research shows the strengths of the SDCE method are:

• Scope of the Evaluation - The SDCE team evaluates an offeror's software development capability against the SDCE model. The SDCE model is composed of

critical capabilities that have historically been high-risk areas in past software developments. The SDCE model address areas not found in the CMM, such as systems engineering, human resources, and specific technologies. As a result, the SDCE has a greater scope of evaluation than the SCE.

• Questionnaire - The SDCE essay-type questionnaire solicits more information than the SEI's maturity questionnaire. Of the two methods, the SDCE is the only evaluation method that may be used in "without discussions" source selections. The process of answering the SDCE questions also provides an offeror insight into its own software development capability. The criteria for scoring the SDCE questions are also defined.

• Project Focus - Both the SDCE and SCE focus on the processes and capabilities required to develop a specific project's software. However, as discussed earlier, because of the SDCE's greater scope, it focuses more on the processes and capabilities required to successfully develop the project's software.

• Site Visit - A strength of the SDCE visit is that it is conducted in a non-adversarial setting where the offeror is given a chance to respond to the SDCE team's findings. This ensures information provided by the offeror was not misinterpreted or lost by the SCE team.

2. SDCE Weaknesses

This research shows the weaknesses of SDCE method are:

• Evaluation Costs - The findings show that a greater volume of information is analyzed prior to the site visits in the SDCE than in the SCE. There is a potential for a SDCE team to be greater in size than that of a SCE team. Unlike the SCE, SDCE core

team members are also members of the SSEB. Their duties in evaluating offerors as members of the SSEB are part of the SDCE. As a result, the cost of conducting a SDCE could be double or triple the cost of conducting a SCE.

• Tailoring - The SDCE model contains critical capabilities that have shown to be high-risk areas for software development. A problem facing SDCE users is limiting the CCs and associated questions for use in a SDCE evaluation to those high-value discriminators that will provide the greatest insight into an offeror's software development capability.

• Process Improvement - The SDCE is strictly an evaluation tool and was not designed to promote process improvement. Unlike the CMM, the SDCE model structure does not help an organization identify and prioritize process improvement activities. The stability of an organization's process improvement plans could be continuously disrupted by the organization's attempt to correct weaknesses found during the most recent SDCE evaluation. This could have a negative impact on promoting overall process improvement in the software industry.

• Unproven Method - The SDCE was developed by combining the strengths and correcting the weaknesses of two proven evaluation methods, the SDCCR and SCE. The model used by the SDCE is a collection of critical capabilities that have historically been areas of high risk in software development. While these two factors may indicate that the SDCE is an effective means of evaluating an offeror's software development capability, the SDCE method is still unproven. It will continue to remain so until the pilot tests of the SDCE are completed next fiscal year.

3. SCE Strengths

The research shows the strengths of the SCE to be:

• Cost of the Evaluation - For reasons cited earlier, the cost of conducting a SCE could be half or one-third the cost of conducting a SDCE.

• Individual Interviews - The SCE team conducts individual interviews during the site visit. Individual interviews verify the information found in an offeror's documentation and evaluate the level of understanding of an organization's software development process by its employees. The SCE's position is that processes not understood by software development personnel are usually not followed. Inconsistencies between the information gathered from documentation and interviews may also indicate a weakness in a process.

• Process Improvement - As discussed previously, the SCE promotes process improvement in the software industry. Using the results of the SCE as a factor in selecting a contractor provides an incentive for the software industry to initiate process improvement in order to stay competitive for Government contracts. The CMM, on which the SCE is based, was designed as a tool to assist an organization in identifying and prioritizing software process improvement activities. A continuous movement toward improving process capability by the software industry will benefit almost all weapon system programs throughout DOD.

• Tailoring - In comparison to the SDCE, the SCE is easier to tailor to a specific acquisition. The SCE method uses the maturity levels of the CMM and target process capability developed by the SCE team for a particular acquisition. The KPAs selected

for use in the evaluation are those attributed to the target process capability and the maturity levels below it. This ease of tailoring the SCE to a specific acquisition makes the method easier to use.

• Proven Method - The SCE and CMM were developed in the late 1980s. The current versions of the SCE and CMM represent approximately eight years of continuous improvement based on lessons learned and feedback from the Government and software industry. Unlike the SDCE, the SCE has a proven history of success.

4. SCE Weaknesses

The research found the following areas to be weaknesses

• Scope of the Evaluation - The SCE team evaluates an offeror's software development capability against the CMM. The CMM is composed of a limited set of key process areas that have been proven to enhance software development and maintenance capability. Unlike the SDCE model, the CMM does not address non-software issues that impact the development of software such as systems engineering, human resources, and specific technology. As a result, the SCE has a smaller scope of evaluation than the SDCE.

• Results Based on Small Samples - The SCE attempts to characterize an organization's software development process capability by evaluating three to four of the organization's past or ongoing projects. Each project has unique characteristics and requirements. The processes of the projects selected for evaluation by the SCE may not be indicative of the entire organization.

• SCE Results Do Not Match the SPA - The SEI acknowledges that SCE results may differ from those of an SPA even though they are both based on the CMM. When this occurs, the validity of the SCE results may be questionable. The SEI has created the CBA project to correct this problem.

C. RECOMMENDATIONS

Based on the strengths and weaknesses, the researcher has developed the following recommendations:

• PMs should select the SDCE method when their only concern is to conduct a thorough evaluation of an offeror's software development capability. This includes the evaluation of non-software issues that might affect the software development project. The overall goal is to reduce the software development risks to his program.

• PMs should select the SDCE method when the SSA directs that the contract be awarded with "without discussions." The SDCE is the only method of the two that permits an evaluation of an offeror's software development capability under "without discussion" conditions.

• PMs should select the SCE method when there is concern with not only conducting an evaluation of an offeror's software development capability, but also promoting process improvement. The overall goal would be to raise the maturity levels of the software industry in order to provide DOD with a mature supplier base capable of developing quality software within time and costs constraints.

• PMs should select the SCE method when their program resource constraints prohibit the funding of a SDCE, but can support the use of a SCE during source selection evaluation.

• When faced with human resources constraints, PMs should select the method which best matches the availability of qualified/experienced personnel to the recommended evaluation team qualifications for the SDCE and SCE.

D. AREAS FOR FURTHER RESEARCH

The researcher encountered the following issues that are important but were outside the scope of this thesis:

• What are the contractor's costs to prepare and participate in a SCE or SDCE? How do these costs affect a contractor's decision on whether or not to bid on a contract?

• How may the SCE or SDCE be used for the purpose of monitoring a contractor's performance?

• How does DOD address the problem of a contractor being subjected to multiple software capability evaluations when bidding on several contracts, most of which incorporate the use of a software capability evaluation during source selection?

• What are the advantages, disadvantages, and legal ramifications of providing SCE or SDCE teams with results of previous software capability evaluations to assist the evaluation team in preparing for a specific site visit?

• If the IDA survey was repeated today, what would be the results?

• What is the correlation between a contractor's maturity level and the results it achieves?

• As a related issue, will an effective software developer have a high maturity level?

• How does a contractor respond to the methods used to evaluate them?

• Is it necessary that all KPAs be met before a given maturity level is realized?

• Should criteria associated with high levels of maturity count in evaluating contractors at a lower target process capability?

E. SUMMARY

The differences in the SDCE and SCE evaluation methods are attributed to the goals of the organizations that created them. ASC/AFMC is committed to assisting PMs in the successful development of weapon systems. The SDCE is designed to identify and reduce development risks for embedded software with little regard for promoting process improvement within the software industry. The SEI's goal is to promote software process improvement. Use of the SCE during source selection evaluation, serves to motivate contractors to pursue CMM-based process improvements. The SCE has a price for promoting software process improvement, the scope of the evaluation is more narrow than that of the SDCE. As the SEI, and ASC/AFMC continuously work to improve their evaluation methods, the gap between identifying and reducing software development risks and promoting software process improvement may close.

LIST OF REFERENCES

1. Defense Systems Management College, ***Mission Critical Computer Resources***, p. 1-1, Defense Systems Management College, 1992.

2. Kitfield, James, "Is Software DoD's Achilles' Heel?", ***Military Forum***, p. 1, National Journal Incorporated, July 1989.

3. Headquarters Air Force Materiel Command, ***AFMC Pamphlet 800-61 Acquisition Management Software Development Capability Evaluation***, p. 1, Department of the Air Force, November 1993.

4. General Accounting Office, ***DOD Embedded Computers - Better Focus on This Technology Could Benefit Billion Dollar Weapons Program***, pp. 10-18, General Accounting Office, April 1990.

5. Software Capability Evaluation Project, ***Software Capability Evaluation (SCE) Version 1.5 Method Description***, p. 16, Software Engineering Institute, July 1993.

6. Osterweil, Leon J., "Why is Process Important?", ***Proceedings of the First International Conference On the Software Process Manufacturing Complex Systems***, p. 5, IEEE Computer Society, October 1991.

7. Paulk, Mark C., and others, "Capability Maturity Model, Version 1.1", ***IEEE Software***, p. 20, IEEE Computer Society, July 1993.

8. Software Engineering Institute, ***The Capability Maturity Model: A Tutorial***, pg. 6-7, Software Engineering Institute, August 1993.

9. Paulk, Mark C., and others, ***CMU/SEI-93-TR-24 Software Engineering Institute, Capability Maturity Model (CMM) Software, Version 1.1***, p. 5.2, Software Engineering Institute, February 1993.

10. Software Capability Evaluation Project, ***CMU/SEI-93-TR-18 Software Capability Evaluation (SCE) Version 1.0 Implementation Guide***, p. 1-2, Software Engineering Institute, July 1993.

11. Bollinger, Terry B., McGowan, Clement, "A Critical Look at Software Capability Evaluations", ***IEEE Software***, pp. 31-32, IEEE Computer Society, July 1991.

12. Boykin, Barbara, Aerospace Industries Association (AIA) Letter to the Software Engineering Institute, p. 1, Aerospace Industries Association, March 17, 1992.

13. Ryer, Michael J., Maher, Austin J., ***AdaJUG Position Paper on Software Engineering Process Improvement and Assessment***, p. 2, Ada Joint Users Group (AdaJUG), February 1992.

14. Humphrey, Watts S., ***CMU/SEI-92-TR-7 Introduction to Software Improvement***, p. 10, Software Engineering Institute, June 1992.

15. Babel, Phil, ***The Software Development Capability Evaluation (SDCE) Introduction Briefing***, p. 2, ASC/AFMC, November 1993.

16. Springsteen, Beth, and others, ***Policy Assessment for the Software Process Maturity Model***, p. 49, Institute for Defense Analysis, August 1992.

17. Defense Systems Management College, ***A Management Guide to Systems Engineering***, p. 1, Defense Systems Management College, 1993.

18. General Accounting Office, ***The Cheyenne Mountain Upgrade***, pp. 1-9, General Accounting Office, 1993.

19. General Accounting Office, ***Submarine Combat System Status of Selected Technical Risks in the BSY-2 Development***, p. 1, General Accounting Office, May 1991.

20. Ruggs, David, "Using a Capability Evaluation to Select a Contractor", ***IEEE Software***, p. 44, IEEE Computer Society, July 1993.

21. Dion, Raymond, "Process Improvement and the Corporate Balance Sheet", ***IEEE Software***, p. 35, IEEE Computer Society, July 1993.

22. Humphrey, Watts, Snyder, Terry R., and Willis, Ronald R., "Software Process Improvement at Hughes Aircraft", pp. 11-23, ***IEEE Software***, IEEE Computer Society, July 1993.

23. Baumert, John., "Promising Common Framework, SEI Combines SCE and SPA Projects", ***IEEE Software***, p. 111, IEEE Computer Society, January 1994.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria VA 22304-6145	2
2.	Library, Code 052 Naval Postgraduate School Monterey CA 93943-5002	2
3.	Defense Logistics Studies Information Exchange U.S. Army Logistics Management Center Fort Lee VA 23801-6043	1
4.	OASA (RDA) ATTN: SARD-ZAC 103 Army Pentagon Washington DC 20310	1
5.	Professor David V. Lamm, Code SM/Lt Department of Systems Management Naval Postgraduate School Monterey CA 93943-5000	5
6.	Professor Martin J. McCaffrey, Code SM/Mf Department of Systems Management Naval Postgraduate School Monterey CA 93943-5000	1
7.	Professor James C. Emery, Code SM/Ey Department of Systems Management Naval Postgraduate School Monterey CA 93943-5000	1
8.	CPT Ferdinand M. Raguindin 13607 Lindendale Road Woodbridge VA 22193	4